NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

IMPACT OF ADOPTING COMMERCIAL PRACTICES IN SOFTWARE DEVELOPMENT AND MAINTENANCE

by

Thomas E. Mullins

March 1995

Thesis Advisor: Martin J. McCaffrey

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE March 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Impact of Adopting Commercial Practices in Software Development and Maintenance	5. FUNDING NUMBERS
6. AUTHOR(S) Thomas E. Mullins	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Modern Army armament systems are becoming increasingly reliant on embedded software. The latest Army version of the self-propelled howitzer, Paladin, includes in its subsystems: an inertial navigation and pointing system, an automatic fire control system, on-board prognostics and diagnostics, and embedded training. All of these subsystems are dependent upon software. The replacement for Paladin, Crusader, will be even more software intensive. The software in Paladin and previous armament systems was developed using military standards. On 29 June 1995, the Secretary of Defense directed the services to change from using military standards to commercial practices. MIL-STD-498, Software Development and Documentation, was approved on 4 November 1995 for interim use for two years. During those two years the military and industry are to develop a commercial replacement for MIL-STD-498. For the two year period, existing commercial software standards are to be used to the maximum extent practicable. This thesis addresses the impact of adopting commercial practices in the development and maintenance of embedded software for Army armament systems. It provides initial insight into the impact on contracting for development and maintenance, test and evaluation, maintenance, potential contractors and risk for embedded armament system software. Paladin, Crusader and Sense and Destroy Armor (SADARM) are used as examples in the study. The thesis makes recommendations to reduce the impact of the change to commercial software practices. The insights developed in this thesis should provide a basis for early evaluation and modification of implementing procedures and guidelines.

14. SUBJECT TERMS Embedded Software, Software Contracting, Paladin, Crusader, SADARM, Software Management, Software Maintenance, Software Risk Management, Commercial Practices	15. NUMBER OF PAGES 98
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)

IMPACT OF ADOPTING COMMERCIAL PRACTICES IN SOFTWARE DEVELOPMENT AND MAINTENANCE

Thomas E. Mullins
Civilian, Department of the Army
B.S., Central State University of Oklahoma, 1971

Submitted in partial fulfillment of the
requirement for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
MARCH 1995

Author: ______________________________
Thomas E. Mullins

Approved by: ______________________________
Martin J. McCaffrey, Primary Adviser

____
Orin E. Marvel, Associate Adviser

______________________________ for
David R. Whipple, Jr., Chairman, Department
of Systems Management

ABSTRACT

Modern Army armament systems are becoming increasingly reliant on embedded software. The latest Army version of the self-propelled howitzer, Paladin, includes in its subsystems: an inertial navigation and pointing system, an automatic fire control system, on-board prognostics and diagnostics, and embedded training. All of these subsystems are dependent upon software. The replacement for Paladin, Crusader, will be even more software intensive. The software in Paladin and previous armament systems was developed using military standards. On 29 June 1995, the Secretary of Defense directed the services to change from using military standards to commercial practices. MIL-STD-498, Software Development and Documentation, was approved on 4 November 1995 for interim use for two years. During those two years the military and industry are to develop a commercial replacement for MIL-STD-498. For the two year period, existing commercial software standards are to be used to the maximum extent practicable. This thesis addresses the impact of adopting commercial practices in the development and maintenance of embedded software for Army armament systems. It provides initial insight into the impact on contracting for development and maintenance, test and evaluation, maintenance, potential contractors and risk for embedded armament system software. Paladin, Crusader and Sense and Destroy Armor (SADARM) are used as examples in the study. The thesis makes recommendations to reduce the impact of the change to commercial software practices. The insights developed in this thesis should provide a basis for early evaluation and modification of implementing procedures and guidelines.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. GENERAL 1

B. RESEARCH QUESTIONS 3

1. Primary Question 3

2. Subsidiary Questions 3

C. SCOPE AND METHODOLOGY OF THE THESIS 4

D. BENEFITS OF THE STUDY 7

E. ORGANIZATION 8

II. BACKGROUND 9

A. DoD SOFTWARE POLICY 9

1. DOD-STD-2176A 9

2. MIL-STD-498 12

3. Specifications and Standards -- A New Way of Doing Business 14

B. ARMY ARMAMENT SYSTEMS 16

1. Paladin 17

2. Crusader 21

3. Sense and Destroy Armor (SADARM) 23

C. SUMMARY 24

III. CONTRACTING FOR SOFTWARE DEVELOPMENT AND MAINTENANCE 27

A. RESULTS OF INTERVIEWS 28

B. DISCUSSION 32

C. SUMMARY 34

IV. MAINTAINABILITY OF SOFTWARE 35

A. RESULTS OF INTERVIEWS 36

B. DISCUSSION 39
C. SUMMARY 42

V. TEST AND EVALUATION OF SOFTWARE 45
A. RESULTS OF INTERVIEWS 47
B. DISCUSSION 49
C. SUMMARY 50

VI. SOFTWARE CONTRACTORS 53
A. RESULTS OF INTERVIEWS 54
B. DISCUSSION 55
C. SUMMARY 57

VII. RISK AND THE POLICY CHANGE 59
A. RESULTS OF INTERVIEWS 63
B. DISCUSSION 64
C. SUMMARY 67

VIII. CONCLUSIONS 69
A. ANSWERS TO RESEARCH QUESTIONS 69
1. First Subsidiary Question 69
2. Second Subsidiary Question 70
3. Third Subsidiary Quesion 71
4. Fourth Subsidiary Question 71
5. Fifth Subsidiary Question 72
6. Primary Research Question 73
B. RECOMMENDATIONS FOR FURTHER RESEARCH 74
1. Commercial Practices in Software Development and Maintenance 74
2. Risks Associated with Using Commercial Practices to Develop Military Software 75

3. IEEE/EIA Standard XXXX 75
4. Cost Reduction and Commercial Software Standards 75
5. Commercial Practices in the Acquisition of Military Hardware 76
C. A FINAL THOUGHT 76

LIST OF REFERENCES 77

LIST OF INTERVIEWS 79

BIBLIOGRAPHY 81

INITIAL DISTRIBUTION LIST 85

I. INTRODUCTION

This thesis investigates the impact of adopting commercial practices in the development and maintenance of embedded software for Army armament systems. The software embedded in today's armament systems was developed using Department of Defense (DoD) and military standards. These standards are grounded in the lessons learned during the development of software for earlier military systems.

A. GENERAL

Not long ago the typical armament weapon system consisted of little more than a gun and a bullet. The more complicated armament systems, such as self-propelled artillery, consisted of a gun, projectile, propellant, optical sight, and an automotive chassis that was originally used by a tank or armored personnel carrier -- little more than a gun, a bullet, some way to point the gun, and a way to move the gun.

Today, the situation has changed. The latest United States Army (US Army) version of the self-propelled howitzer, the Paladin, includes in its subsystems: a ring laser inertial navigation and pointing system; embedded training for the crew and maintainers; and an automatic fire control system (AFCS). The AFCS receives a digital request for fire, selects the optimum ammunition, computes the aiming data, aims the gun, and makes all of the relevant communications to outside agencies. All of these subsystems are dependent upon software. The AFCS software alone comprises in excess of one million lines of Ada code. Development of the software for the Paladin howitzer consumed over one third of the resources required to develop this latest product improvement to the venerable M109-series, 155 millimeter, self-propelled howitzer. [Ref. 1]

The Paladin howitzer demonstrates the increasing importance of embedded software in modern armament weapon systems. The software embedded in today's armament systems was developed using government specific standards. These standards specify the process for developing the software, the format and content of software documentation, and the review system for approval of the software and its documentation.

Software development standards were developed as a result of early software development experience. They are designed to reduce risk by insuring that software will perform the task required, be repairable when problems arise, and be modifiable (maintainable) as requirements change in the future. Department of Defense Standard 2167A (DOD-STD-2167A), Military Standard Defense System Software Development, was the software development document in force during the most important phases of Paladin development. Early Paladin development was initiated under DOD-STD-1679, Defense System Software Development.

Software development for armament systems requires the selected developmental contractors to conform to appropriate DoD and military software and systems engineering standards. DOD-STD-2167A has been the baseline standard for development of embedded software since February 1988. MIL-STD-498, System Software Development and Documentation, was recently approved for use on 8 November 1994 as a replacement for DOD-STD-2167A.

A recent change in DoD policy will modify the software development process. Future software development contracts will no longer require the use of DOD-STD-2167A or other related DoD or military software and systems engineering standards. However, the new policy does not prevent a contractor from proposing and using a DoD or military standard to develop armament system software. The new policy does prevent Program Managers (PMs) and other government managers of software development and maintenance programs from requiring contractors to use MIL-STD-498 and/or other related DoD and military standards for future procurements. Government managers are now required to permit contractors to propose and use commercially accepted standards, processes, procedures and practices. A waiver allowing the requirement of these standards may be granted if their use provides a substantial economic advantage to the government. Army waivers are approved by the milestone decision authorities or the Army Acquistion Executive (AAE). [Ref. 2]

Government managers of armament software development programs have come to rely on the provisions of DOD-STD-2167A and related DoD and military standards to control development risk by including requirements' traceability, maintainability, efficiency and effectiveness issues into the software development process. As previously mentioned, the DoD and military standards prescribing software development policy evolved from the lessons learned in previous software development programs. In a sense, government software specifications came into being as a result of the inadequacy of early commercial software development practices. The new DoD policy requires that government managers allow the use of any commercially accepted practice that meets government requirements in the development of embedded armament software [Ref. 2].

B. RESEARCH QUESTIONS

This research seeks to answer several questions regarding this recent policy change.

1. Primary Question

The primary question for this thesis is: What is the effect on armament system acquisitions of allowing commercial practices to be used instead of requiring DoD or military standards in the development and maintenance of embedded software?

2. Subsidiary Questions

Contained within this critical issue are five additional issues that lead to the following questions:

- How will this change in policy affect the way PMs and other government managers contract for software development and maintenance?
- How will this change in policy impact the maintainability of armament software?
- How will this change in policy influence the test and evaluation of armament software?
- How will this change in policy affect potential government contractors for armament system software?
- Will this change in policy affect risk in the development and maintenance of armament systems?

C. SCOPE AND METHODOLOGY OF THE THESIS

The objective of this thesis is to provide initial insight into the issue of returning to commercial practices for the development of armament system software. This thesis should provide the basis for identifying key concerns in implementing policy. By identifying the concerns early in the implementation of policy, government managers can take steps to mitigate potential adverse effects and reinforce the positive results of the policy change.

The key domains for answering these questions lies in four communities: (1) program management, (2) life cycle software support, (3) test and evaluation, and the (4) armament software contractor.

The primary research question will be addressed through the five subsidiary questions. Three armament programs will be used as the principal vehicles to focus the investigation of these issues -- the Paladin, Crusader, and Sense and Destroy Armor (SADARM) programs.

Paladin, Crusader and SADARM have each established program management organizations. They are managed for the Army by the Program Executive Office for Field Artillery Systems (PEO FAS -- formerly PEO Armaments). The PEO FAS organization is illustrated in Figure 1.

The Paladin program is the most recently completed, software intense, major armament development program. Paladin software was developed using DOD-STD-1679 and DOD-STD-2167A. Paladin is currently in fielding and the software is already in the process of being updated to accommodate changes in the fire support arena.

Crusader is the next planned major armament program. Crusader, until recently, was named the Advanced Field Artillery System (AFAS). Crusader will develop a new automated 155 millimeter howitzer to replace the M109-series self-propelled howitzers (including Paladin). Crusader has just completed the Defense Acquisition Board (DAB) process for Milestone I and is entering into the Demonstration and Validation (DEM/VAL) phase of development. Crusader is projected to be a software intensive

system. It will develop its software under the new policy of commercial practices and/or MIL-STD-498. [Ref. 1]

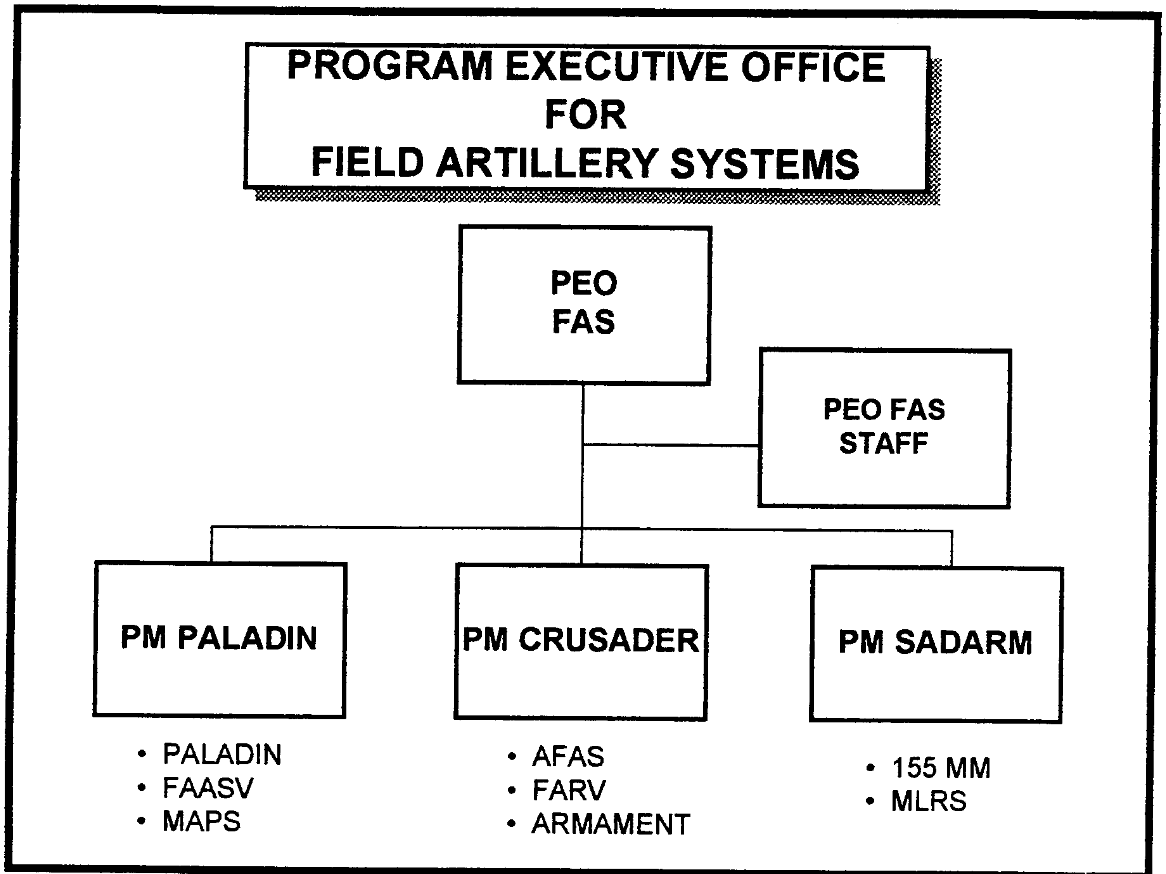

Figure 1. The Program Executive Office for Field Artillery Systems (PEO FAS) Organization

SADARM is a smart munition program. SADARM is developing smart sub-munitions for employment by 155 millimeter cannons and the Multiple Launch Rocket System (MLRS). The 155 millimeter sub-munition has been approved for production. The MLRS sub-munition is in the Engineering and Manufacturing Development (EMD) phase of development.

The first subsidiary question (How will this change in policy affect the way PMs and other government managers contract for software development and maintenance?) was investigated by interviewing PEO FAS, PM Paladin, PM SADARM, PM Crusader, the Chief of the Life Cycle Software Support Center (LCSSC) at Picatinnay Arsenal, and

their staffs. The PEO and the PMs are responsible for the development and maintenance of the software for their systems prior to the completion of fielding. The LCSSC is the principal agency for support of embedded software in armament systems once a system is fielded.

The second subsidiary question (How will this change in policy impact the maintainability of armament software?) was investigated primarily through interviewing members of the Picatinny LCSSC. The LCSSC is responsible for maintenance of Paladin software and for insuring Crusader software is maintainable. Additionally, this subject was investigated during interviews with program management personnel and contractor software personnel.

The third subsidiary question (How will this change in policy influence the test and evaluation of armament software?) was answered by interviewing members of the Test Integration Working Groups (TIWG) for SADARM, Crusader and Paladin. TIWG members interviewed for this question represented the operational and developmental testers and independent evaluators.

The fourth subsidiary question (How will this change in policy affect potential government contractors for armament system software?) was pursued through interviews with the software contractors for Paladin, Crusader and SADARM. The investigation of this issue was limited because of current government solicitations for the development of Crusader software and the maintenance of Paladin software. Contractors were sensitive to questions.

The last subsidiary question (Will this change in policy affect risk in the development and maintenance of armament systems?) was addressed through interviews with members of all four of the communities. Answering this question required the researcher to analyze the results of interviews and synthesize a consensus response.

A considerable amount of literature has been published on the subjects of software development, software maintenance and the use of DOD-STD-2167A. However, at this point no articles have been found dealing specifically with application of the new DoD

policy on the use of commercial practices in software development. With any major policy change, there is a period of time when details of the implementation actions and impact of the policy are uncertain. This general lack of information within the DoD software community on the new DoD policy severely limits the effectiveness of using a questionnaire to gather data on the thesis topic. As a result, no questionnaire or survey was distributed.

The lack of literature and uncertainty of information on the new policy lead to conducting the investigation by interview. To offset the general lack of knowledge on the subject, interviews were conducted in two stages. The first stage of an interview in many cases consisted of educating the person interviewed. The individual was provided with general subjects and questions to be covered in the interview. The second phase of the interview process consisted of a traditional interview structured around the questions and subjects provided during the first phase. The results of the interviews were analyzed by comparing and contrasting the opinions of the experts interviewed. Areas of consensus and disagreement were used to establish trends and areas for additional analysis.

The uncertainty of information on the new DoD policy and time constraints also led to limiting the scope of the investigation to three programs -- Paladin, SADARM and Crusader.

D. BENEFITS OF THE STUDY

This analysis on the effects of procuring and maintaining armament software through commercial practices should provide insight into the initial implementation of the policy change. The information developed in this thesis should provide a basis for early evaluation and modification, if necessary, of implementation procedures for using commercial practices to develop embedded armament system software. By identifying concerns early in the implementation of policy, steps may be taken to mitigate potential adverse effects and reinforce the positive results of this policy change. Additionally, this analysis can provide the basis for future investigation into the process after the policy has been more widely implemented.

E. ORGANIZATION

This thesis consists of eight chapters. Chapter II establishes the background for investigating the problem. It provides a brief description of DOD-STD-2167A and MIL-STD-498. It provides insight into the policy change from DoD and military standards to commercial practices. Additionally, Chapter II briefly describes the Paladin, Crusader and SADARM programs. Chapter III begins the investigation by presenting the results of the first subsidiary question. It addresses how this change in policy will affect the way PMs and other government managers contract for software development and maintenance?

Chapters IV through VII present the results of the second through fifth subsidiary questions. The results of the interviews and an analysis of the issues raised during the interviews are presented for each subsidiary question. Chapter VIII provides the answers to the primary and subsidiary research questions. This chapter summarizes the issues discussed in previous chapters and uses the results to draw conclusions and make recommendations. The chapter concludes with recommendations for further research.

II. BACKGROUND

This chapter establishes the framework used in the investigation of the thesis questions. The chapter is divided into two sections. The first section looks at DoD software policy. It provides a brief description of DOD-STD-2167A and MIL-STD-498. Additionally, it provides a discussion into the policy change from DoD and military standards to commercial practices.

The second section provides a brief description of the three Army armament programs used for this study. The programs are Paladin, Crusader and Sense and Destroy Armor (SADARM). All three programs involve significant amounts of embedded software. The contractor and government personnel interviewed in the conduct of the study are currently or were involved in the development, maintenance, test and evaluation, or oversight of the software in these programs.

A. DoD SOFTWARE POLICY

Traditionally, Army armament system embedded software is developed using military standards. DOD-STD-2167A has been the basic standard for development of military embedded software since 1988 [Ref. 3]. Recently, on 8 November 1994, MIL-STD-498 was approved as a temporary replacement for DOD-STD-2167A [Ref. 4]. This temporary approval of MIL-STD-498 was in response to the Secretary of Defense's (SECDEF) initiative to move from military standards to commercial practices [Ref. 2]. During the next two years, DoD and industry are to develop a joint standard to replace MIL-STD-498 [Ref. 4]. An understanding of DOD-STD-2167A, MIL-STD-498 and the SECDEF's policy memorandum of 29 June 1994 is necessary to assess the impact of change in software development.

1. DOD-STD-2167A

DOD-STD-2167A is a process standard for the development of embedded software. It defines the software development process for military embedded software. The

standard does not specify the development method or model to be used. However, it does set process and documentation standards that the selected developmental model must support. The standard is designed to be tailored to specific software developmental models and products. DOD-STD-2167A applies throughout the life cycle of embedded software. [Ref. 3]

The software development process is divided into major activities or phases. The major activities specified by DOD-STD-2167A are:

- Systems Requirements Analysis/Design.
- Software Requirements Analysis.
- Preliminary Design.
- Detailed Design.
- Coding and Computer Software Unit (CSU) Testing.
- Computer Software Component (CSC) Integration and Testing.
- Computer Software Configuration Item (CSCI) Testing.
- System Integration and Testing. [Ref. 3]

Each major activity or phase terminates in a formal review or audit. Additionally, each major activity or phase produces associated documentation. The relationship among the major activities or phases, documentation and formal reviews or audits is illustrated in Figure 2. [Ref. 3]

A major function of software documentation is to trace software requirements through the development process. The documentation ties software requirement objectives with standards for performance, software design, test plans, software code, and the results of software test and evaluation. This linkage is essential and enhances the ability to perform maintenance and troubleshooting of software. Additionally, documentation assists in ensuring all required actions in each phase of software documentation are adequately performed. The format for documentation is contained in the Data Item Descriptions (DIDs) associated with DOD-STD-2167A. The format of DIDs may be tailored (modified) to meet individual program requirements. [Ref. 5]

Software reviews and audits ensure that all actions and documentation for each phase of development are adequately performed. The conduct of formal reviews and audits is contained in MIL-STD-1521, Progrm Reviews. [Ref. 3]

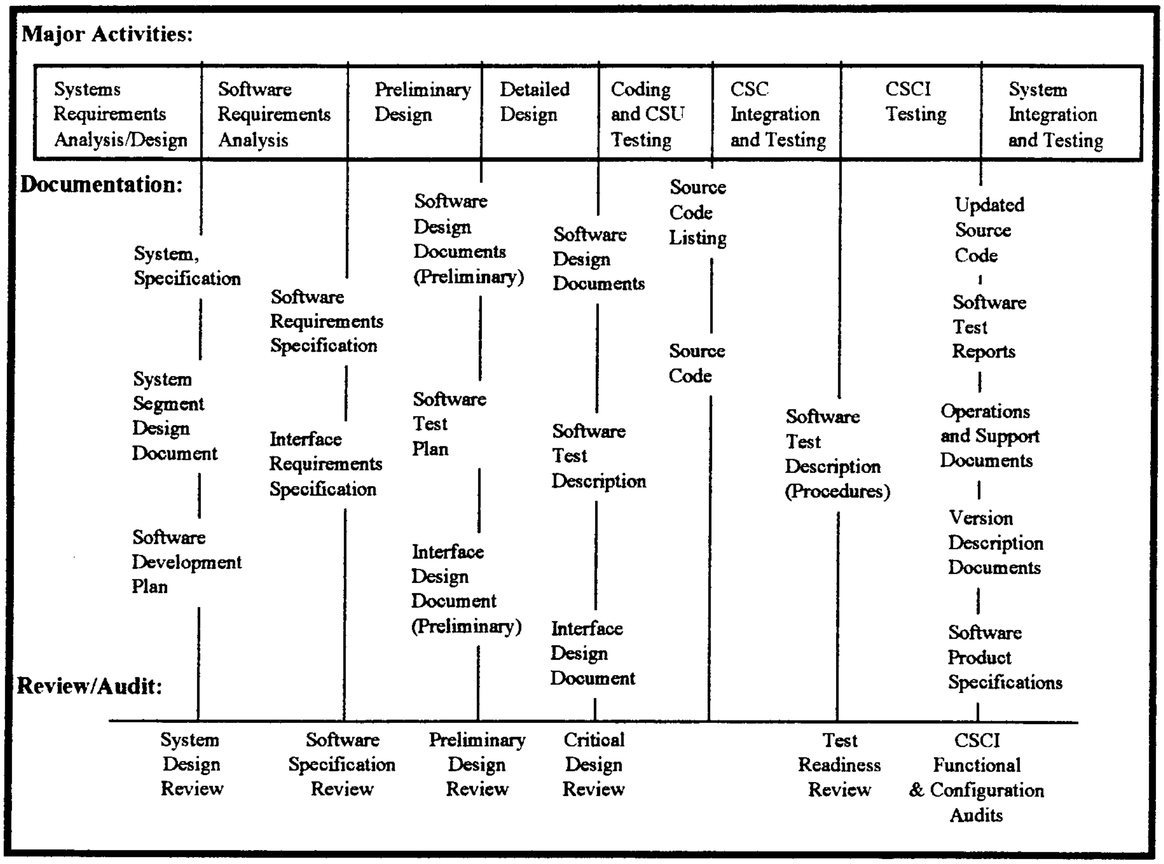

Figure 2. Software Development Phases, Documentation, Reviews and Audits

Execution of the major activities or phases of software development in the order they are presented in Figure 2 results in the stepwise or "waterfall" model for software development. In the "waterfall" model each phase is completed, documented, and reviewed or audited prior to moving to the next phase of development. If actions in the current phase affect results of the previous phase, the documentation for the previous phase is updated. The new documentation and updated documentation from the previous phase are approved during the current phase review or audit. This was the most prevelant model used for DoD software development in the 1970s and 1980s. [Ref. 3, 5 and 6]

Software development models other than "waterfall" are supported by tailoring the development process. The major activities or phases of software development may be overlapped, applied iteratively or recursively to modify or tailor the process. Also, additional actions or phases, such as prototyping, may be added. [Ref. 3]

Some examples of other software development models are rapid prototyping, evolutionary and spiral. Rapid prototyping models introduce the use of software prototypes. Evolutionary models iterate through the development process until a satisfactory product is developed. The spiral model develops a hypothesis, tests the hypothesis, and modifies the hypothesis as required by test results. When the hypothesis develops a satisfactory solution to a development phase, the development proceeds into the next development phase. [Ref. 6]

Since the implementation of DOD-STD-2167A in February 1988, the "state-of-the-art" in software development has continued to progress and the military acquisition environment has changed. The Object Oriented Development (OOD) process is gaining acceptance in the systems engineering of software intensive systems. The emergence of Computer-Aided Software Engineering (CASE) tools are introducing automation into the development of embedded software. More recently, DoD is emphasizing the concept of reusable software as a means of reducing the cost of software development. Additionally, revision of the DoD 5000 series of regulations and instructions has altered much of the acquisition environment in general. [Ref. 7]

2. MIL-STD-498

MIL-STD-498 replaced DOD-STD-2167A and DOD-STD-7935A, Military Standard Automated Information Systems (AIS), on 8 November 1994 [Ref. 8]. It is the result of the evolution in software development. It makes no drastic departures from the process described in DOD-STD-2167A. The foreword to MIL-STD-498 summarizes the changes as follows:

> This standard merges DOD-STD-2167A and DOD-STD-7935A to define a set of activities and documentation suitable for the development of both weapon systems and Automated Information Systems. A conversion guide from these standards to MIL-STD-498 is provided ... Other changes include improved compatibility with non-hierarchical design methods (OOD); improved compatibility with computer-aided software engineering (CASE) tools; alternatives to, and more flexibility in preparing documents; clearer requirements for incorporating reusable software; introduction of

software management indicators (metrics); added emphasis on software supportability; and improved links to systems engineering. [Ref. 8]

The "waterfall" development model is de-emphasized in MIL-STD-498. The new standard allows greater flexibility in tailoring to meet software development models. Additionally, MIL-STD-498 provides specific guidance on tailoring. It includes examples on tailoring to meet grand design, incremental and evolutionary models. [Ref. 8]

Unlike DOD-STD-2167A, MIL-STD-498 does not require the use of other DoD or military standards. Processes for configuration management, product review and audit, quality control, risk management, and security are now contained in the new standard. It permits greater flexibility than MIL-STD-1521 in reviews and audits by allowing them to be tailored to individual software development effort requirements. In addition to providing a backward compatible reference to DOD-STD-2167A and DOD-STD-7935A, MIL-STD-498 provides a reference relating major development activities to recognized commercial industry standards. [Ref. 8]

The DIDs for the new standard specify content, encourage the use of compatible commercial items that meet contract requirements, and do not specify the format for documentation. Twenty two DIDs are included with MIL-STD-498. Only the DIDs required to support a particular development are specified. Electronic media is allowed to replace paper documentation. [Ref. 7 and 8]

Fourteen of these DIDs were common to both of the merged standards. They are:

- Software Development Plan (SDP).
- Software Transition Plan (STrP), formerly a portion of the Computer Resources Integrated Support Document (CRISD).
- System/Segment Specification (SSS).
- System/Segment Design Document (SSDD).
- Operational Concept Document (OCD), formerly a portion of the SSS.
- System Requirements Specification (SRS).
- Interface Requirements Specification (IRS).
- Software Design Document (SDD).
- Interface Design Document (IDD).
- Software Test Plan (STP).
- Software Test Description (STD).

- Software Test Report (STR).
- Software Product Specification (SPS).
- Software Users Manual (SUM). [Ref. 7 and 8]

Four of the document requirements are carried forward from the AIS standard. The DIDs are:

- Software Installation Plan (SIP).
- Data Base Design Description (DBDD).
- Software Center Operator Manual (SCOM). AIS systems only.
- Software Input/Output Manual (SIOM). AIS systems only. [Ref. 7 and 8]

Four are from DOD-STD-2167A only. The documents are:

- Computer Programming Manual (CPM), formerly called Software Programmer's Manual (SPM).
- Computer Operation Manual (COM), formerly called Computer System Operator's Manual (CSOM).
- Firmware Support Manual (FSM).
- Version Description Document (VDD). [Ref. 7 and 8]

Software development standards are brought into full compliance with other acquisition regulations and instructions. Management of software development is simplified by not requiring the use of other DoD or military standards -- MIL-STD-498 provides "one-stop shopping" for software management. Additionally, by providing a cross-reference between itself and existing commercial standards, MIL-STD-498 reduces the gap between military and commercial software development.

3. Specifications and Standards -- A New Way of Doing Business

William J. Perry, the Secretary of Defense, formalized the movement from military standards and specifications to civilian standards and performance specifications with his 29 June 1994 memorandum, "Specifications and Standards -- A New Way of Doing Business." This memorandum directs the implementation of the recommendations from a Process Action Team (PAT) formed by the Under Secretary of Defense for Acquisition Reform {USD(AR)}. The USD(AR) chartered the PAT to "develop a strategy and plan of action to decrease reliance, to the maximum extent practicable, on military specifica-

tions and standards." Mr. Darold Griffin, Deputy to the Commander of the Army Materiel Command (AMC), led the team. The team's recommendations are in its report titled, "Blueprint for Change." [Ref. 2]

The memorandum states the goal for this change from military standards and specifications as:

> To meet future needs, the Department of Defense must increase access to commercial state-of-the-art technology and must facilitate the adoption by its suppliers of business processes characteristic of world class suppliers. In addition, integration of commercial and military development and manufacturing facilitates the development of dual-use processes and products and contributes to an expanded industrial base that is capable of meeting defense needs at lower costs. [Ref. 2]

The SECDEF's memorandum applies to all military developments and acquisitions including Army embedded software. The service acquisition executive may exempt on-going solicitations or contract negotiations, by waiver, for the next 180 days. The memorandum applies to programs of all Acquisition Categories (ACAT). The memorandum establishes a two-year transition period. DoD Instruction 5000.2, the Defense Federal Acquisition Regulation Supplement (DFARS) and other appropriate policy documents will be changed to effect the policy change. [Ref. 2]

The policy change for military specifications and standards, as stated in the memorandum, is:

> Performance specifications shall be used when purchasing new systems, major modifications, upgrades to new systems, and nondevelopmental and commercial items, for programs in any acquisition category. If it is not practicable to use a performance specification, a non-government standard shall be used. Since there will be cases when military specifications are needed to define an exact design solution because there is no acceptable non-governmental standard or because the use of a performance specification or non-government standard is not cost effective, the use of military specifications and standards is authorized as a last resort, with an appropriate waiver. [Ref. 2]

Milestone decision authorities approve waivers for the use of military standards and specifications. [Ref. 2]

Implementation of the policy change requires DoD to form partnerships with industry to develop or modify commercial standards to replace military standards. During the transition period, contractors and potential contractors may propose the use of any commercially accepted practice or standard as a replacement for the provisions of military standards and specifications. Additionally, contract managers are encouraged to modify existing contracts to eliminate or reduce the use of military standards and specifications through no-cost contract modification. [Ref. 2]

The Institute of Electrical and Electronics Engineers (IEEE) and the Electronic Industries Association (EIA) are working together to develop a commercial standard to replace MIL-STD-498. Approval of the new commercial standard is expected by late 1996. [Ref. 7]

The thesis questions investigate the effect of this policy change on the development and maintenance of embedded software for Army armament systems. The implementation of this policy change is evolving at this time. With any major policy change, there is a period of time when the details of the implementation actions and impact are uncertain. In the case of this thesis, this uncertainty resulted in conducting the investigation of the research questions by interview.

B. ARMY ARMAMENT SYSTEMS

The government and contractor persons interviewed during this investigation are or were associated with one or more of the three Army armament programs presented in this section and are recognized experts in the software development process. The Paladin, Crusader and Sense and Destroy Armor (SADARM) programs are briefly described in this section. All three of the programs involve significant amounts of embedded software. The programs selected cover the acquisition development spectrum from Demonstration and Validation (DEM/VAL), through the Engineering and Manufacturing Development (EMD), and into Production.

1. Paladin

The M109 series howitzer is the most commonly used self-propelled howitzer in the world today. The United States (US), all of its major allies except France, many neutral countries (including Switzerland and Austria), and many former allies (including Jordan, Iraq, Iran, and Vietnam -- acquired with the fall of South Vietnam) use it. All of the combatants in Desert Storm, except France and Syria, employed the M109 howitzer. The US has over 2,400 M109 series howitzers in service today. Over 5,000 M109 series howitzers are in service in the other countries. The M109 series Howitzer has been in continuous production since 1961. [Ref. 9]

The M109A6, Paladin, Howitzer is the latest variant of the venerable M109, Self-Propelled, Howitzer. The Paladin is the product of the Howitzer Improvement Program (HIP) and was known as the HIP howitzer prior to 1989. The Paladin was developed by inserting recent technology from several disciplines into a thirty year old howitzer. The Paladin improves the survivability, responsiveness, reliability, and range of the M109A2/3 Howitzer. However, the revolutionary change introduced by Paladin is not just the incremental change from the insertion of technology in these areas. The main innovation of Paladin is that by combining results of the incremental improvements the howitzer can employ the radical new doctrine of semiautonomous operations, "shoot and scoot tactics." [Ref. 10]

The difference between the tactics of Paladin and previous self-propelled howitzers is at the platoon level. The older tactics are commonly called 3 X 8 operations. A basic understanding of 3 X 8 operations makes understanding semiautonomous operations easier.

The 3 X 8 battery (Figure 3) is divided into two platoons with four howitzers and a Fire Direction Center (FDC) each. Each platoon occupies an area approximately 200 to 300 meters by 150 to 200 meters. The howitzers occupy the front of the position in a lazy W. The FDC locates to the center rear of the position. The platoons are about one to

two kilometers (Km) apart. Internal platoon communications are by land line. External platoon communications are by radio. [Ref. 11]

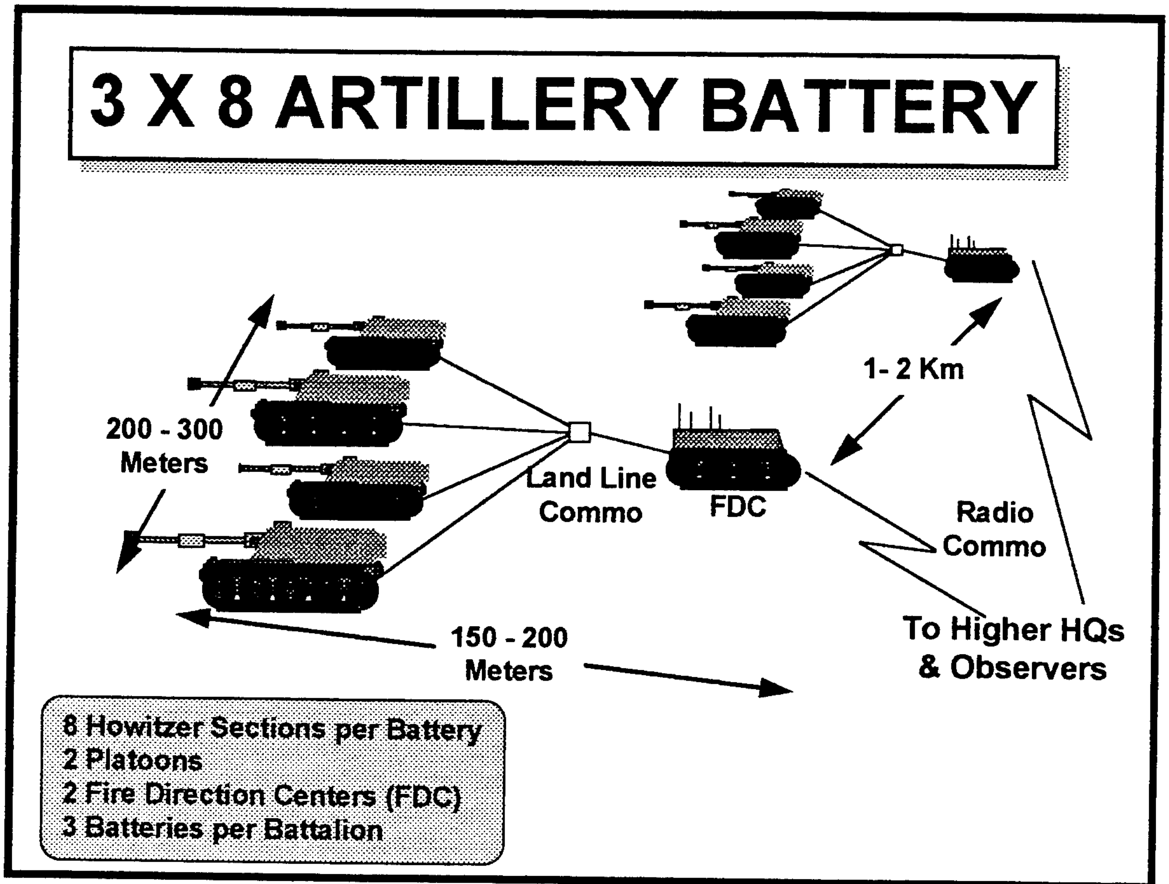

Figure 3. The 3 X 8 Battery

Prior to occupation of the position, the center of the gun line and survey direction are established by the battalion's survey section. Upon occupation, the position of each howitzer and common pointing (lay for direction) is passed to each howitzer with an Aiming Circle (survey instrument). The howitzer section and communication section personnel emplace the land lines. The FDC initializes the Battery Computer System (BCS) with the position data of the howitzers. When these actions are complete the battery is ready to accept requests for fire (fire missions -- targets to attack). Actions necessary to set up take from five to ten minutes to accomplish under emergency conditions, "hip shoot," by a well-trained battery. Under normal conditions these actions take from 20 to 30 minutes. [Ref. 11]

The Paladin battery operates with tactics derived from the 3 X 8 battery. The major difference is within the howitzer platoon (Figure 4). Because of embedded systems the Paladin howitzer is capable of accurate, immediate, self-location and laying. It communicates with the platoon FDC via digital long-range radio. The Paladin does its

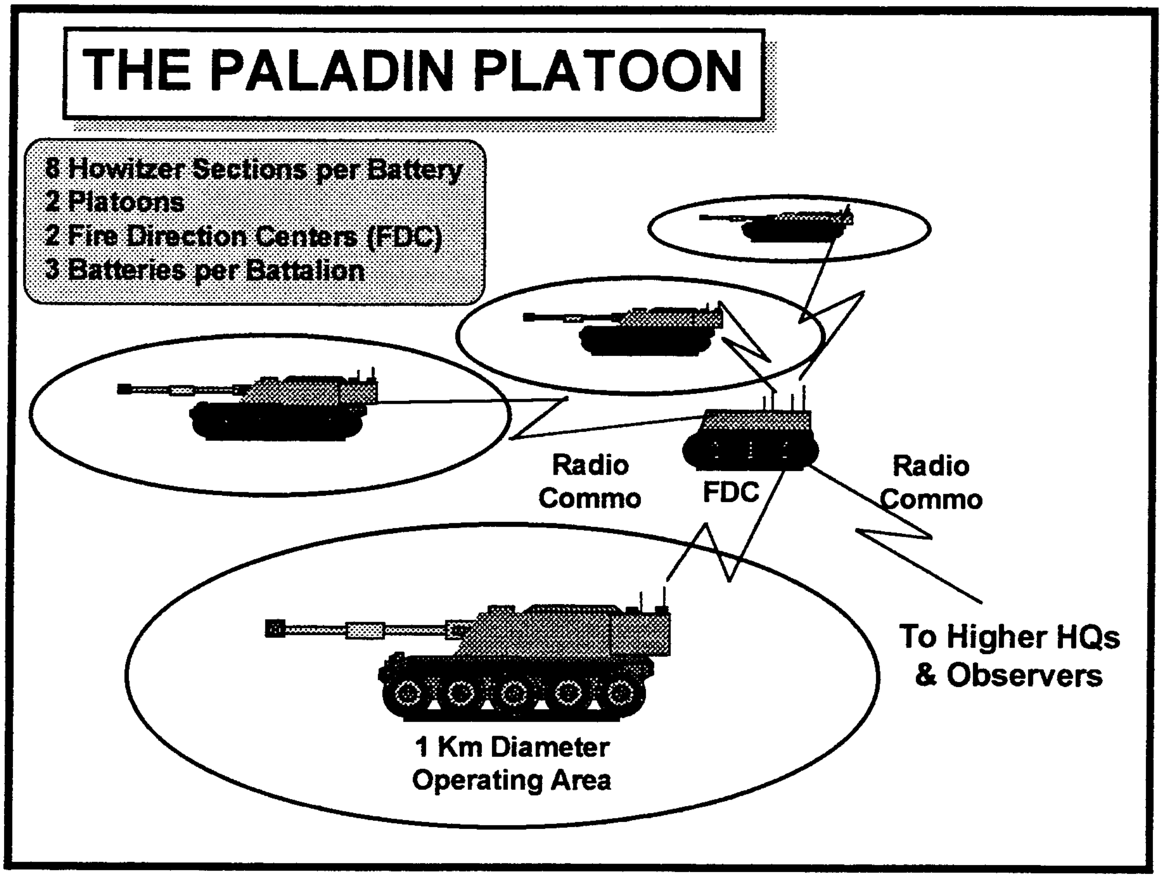

Figure 4. The Paladin Platoon

own, on-board, computation of firing data. A Paladin howitzer is capable of firing from the move within 60 seconds of receiving a fire mission. This allows the Paladin to execute semiautonomous operations. The Paladin moves into a position, waits, then fires when directed by the FDC, and moves to another position. This allows the Paladin to avoid counterfire (attack by enemy artillery). Paladin howitzers are assigned operating areas of about one kilometer (Km) in diameter. Within this area, the Paladin is free to move as required to avoid counterfire. The FDC is located away from the howitzers to avoid attack and optimize communications. [Ref. 12]

Paladin's improvements are summarized in Figure 5. The Automatic Fire Control System (AFCS) and on-board navigation and laying system (MAPS -- Modular Azimuth Positioning System) are each software intensive systems. The AFCS consists of a distributed computer system connected with a 1553B data bus and its software. The MAPS with its software was procured as a "black box" (i.e., it had a form, fit and function specification). [Ref. 13]

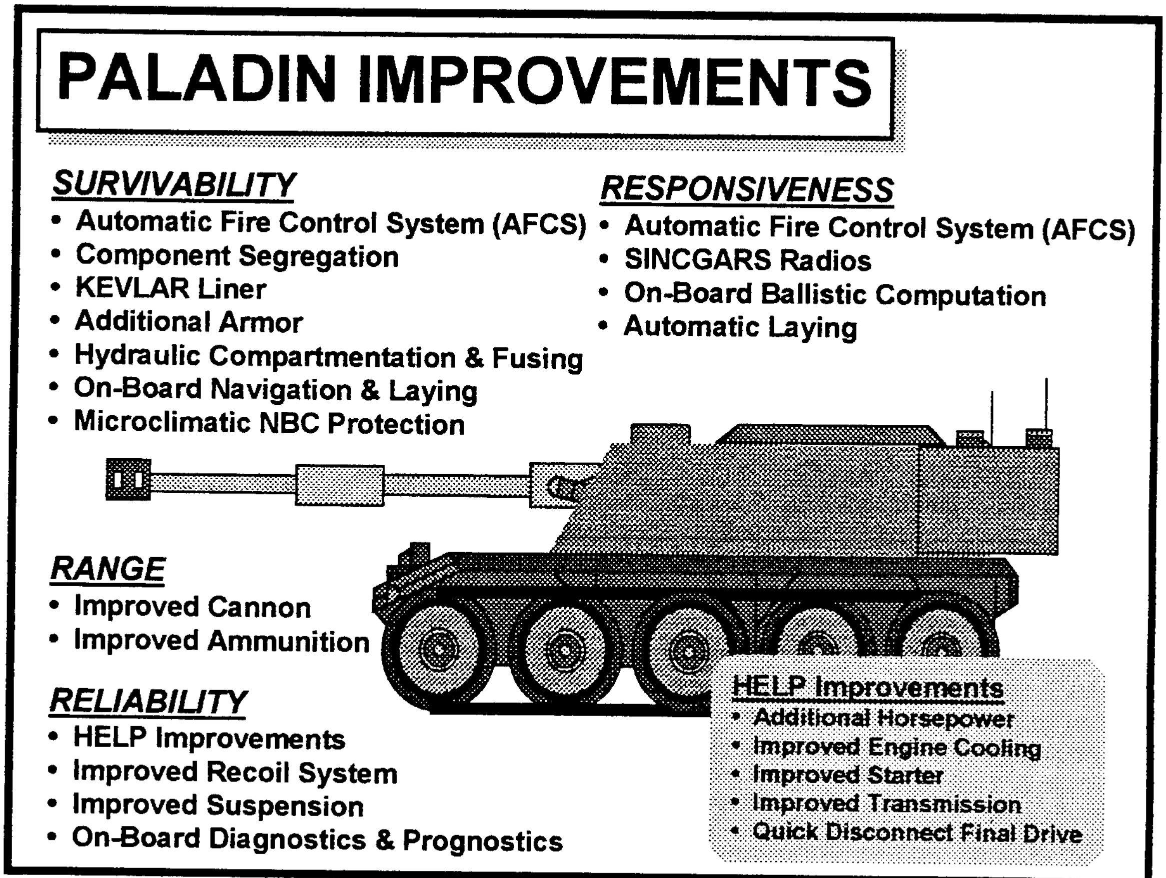

Figure 5. Paladin Improvements

The AFCS software development for Paladin began under DOD-STD-1679 and completed under DOD-STD-2167A. The software is all in the Ada language. It exceeds one million lines of code. The software controls digital communications, graphical interface with the crew, ammunition selection (rule-based), ballistic computation, automatic laying of the cannon, sheafing (rule-based determination of pattern for rounds fired to optimize effects on target), ammunition accountability, automatic determination and

application of gunnery constants, automatic status reporting to FDC, on-board diagnostics, on-board prognostics, and embedded training based upon on-board scenario generation. It accomplishes these tasks in real or near-real time. [Ref. 13]

The AFCS software has been revised twice since fielding began in 1991. The revisions accommodated new munitions and improvements to systems that interface with Paladin. Revision three is completing development now. Additionally, AFCS software is currently being modified to support a processor upgrade and integration of a laser firing system. The MAPS software is being upgraded to integrate the Global Positioning System (GPS). GPS integration removes the requirement for occasional survey support to maintain accuracy. [Ref. 1]

Alliant Techsystems developed the AFCS software with the exception of the diagnostics, prognostics and embedded training software. General Electric (GE) developed the diagnostics and prognostics software. Educational Communications Corporation (ECC) developed the on-board training and institutional trainers and software. Honeywell provides the MAPS. BMY was the system integrator. [Ref. 1]

Paladin fielding is in progress and will complete in early 1999. According to the program office, the Paladin software is among the most trouble-free and easy to maintain of any embedded software of comparable size. [Ref. 1]

2. Crusader

The Crusader program is developing the replacement for Paladin and its' accompanying ammunition resupply vehicle, the Field Artillery Ammunition Support Vehicle (FAASV). Crusader consists of two vehicles. The vehicles are the Advanced Field Artillery System (AFAS) and the Future Armored Resupply Vehicle (FARV). The AFAS is a new 155 millimeter, self-propelled, howitzer. The FARV resupplies AFAS with ammunition, fuel and all other support necessary to keep AFAS in combat. Crusader will exploit technology to dramatically improve effectiveness and efficiency when compared to

the Paladin and FAASV combination. Crusader is in the DEM/VAL phase of development. [Ref. 14]

Crusader will improve effectiveness by improving rate-of-fire, range, flexibility and execution of shoot and scoot tactics. Rate-of-fire will be increased by automating ammunition handling, loading and firing. Range and flexibility will be improved through the use of advanced propellant. Key technologies are Liquid Propellant (LP), automation and robotics. Flexibility and execution of tactics will be enhanced through improved situation awareness and mission processing. Sensors, data transmission and advanced data processing are the crucial technologies required. Common to all of these technologies is the extensive use of embedded software. [Ref. 14]

The key to improved efficiency for Crusader is reducing personnel. Automation, robotics and computer assisted decision making are the critical technologies that provide this savings. Each requires extensive use of embedded software. [Ref. 14]

Crusader software will meter LP to control the range. Software will control the flow of ammunition from ordering through delivery on target. It will advise the crew of the tactical environment, best route to the next position, and status of the system. Crusader will be defined by its' software. [Ref. 14]

The request for proposal for development of Crusader software was based upon a performance specification. The performance specification did not require the use of any DoD or military standards. United Defense is the prime contractor for Crusader development. Crusader software development is split between United Defense and it's subcontractor, Magnavox. Both contractors elected to tailor DOD-STD-2167A and are transitioning to MIL-STD-498. [Ref. 15]

Principal software products from DEM/VAL will be a detailed Software Development Plan (SDP), software metrics requirements, Software Requirements Specification (SRS), reusable software and a software reuse plan. Both software contractors are using the spiral model with rapid prototyping. [Ref. 15]

3. Sense and Destroy Armor (SADARM)

Sense and Destroy Armor (SADARM) is a smart submunition. SADARM is being packaged for delivery by 155 millimeter cannons and the Multiple Launch Rocket System (MLRS). Two submunitions are in each 155 millimeter projectile. An MLRS missile carries six slightly larger submunitions. The 155 millimeter version of SADARM is entering production. MLRS SADARM is nearing the end of EMD. [Ref. 16]

Both versions of SADARM function in the same manner (Figure 6). The SADARM carrier is launched at its intended target. The submunitions deploy from the carrier over the target. A parachute deploys and slows the descent. The infrared and millimeter wave sensors begin scanning for armored vehicles in the target area. Submunition

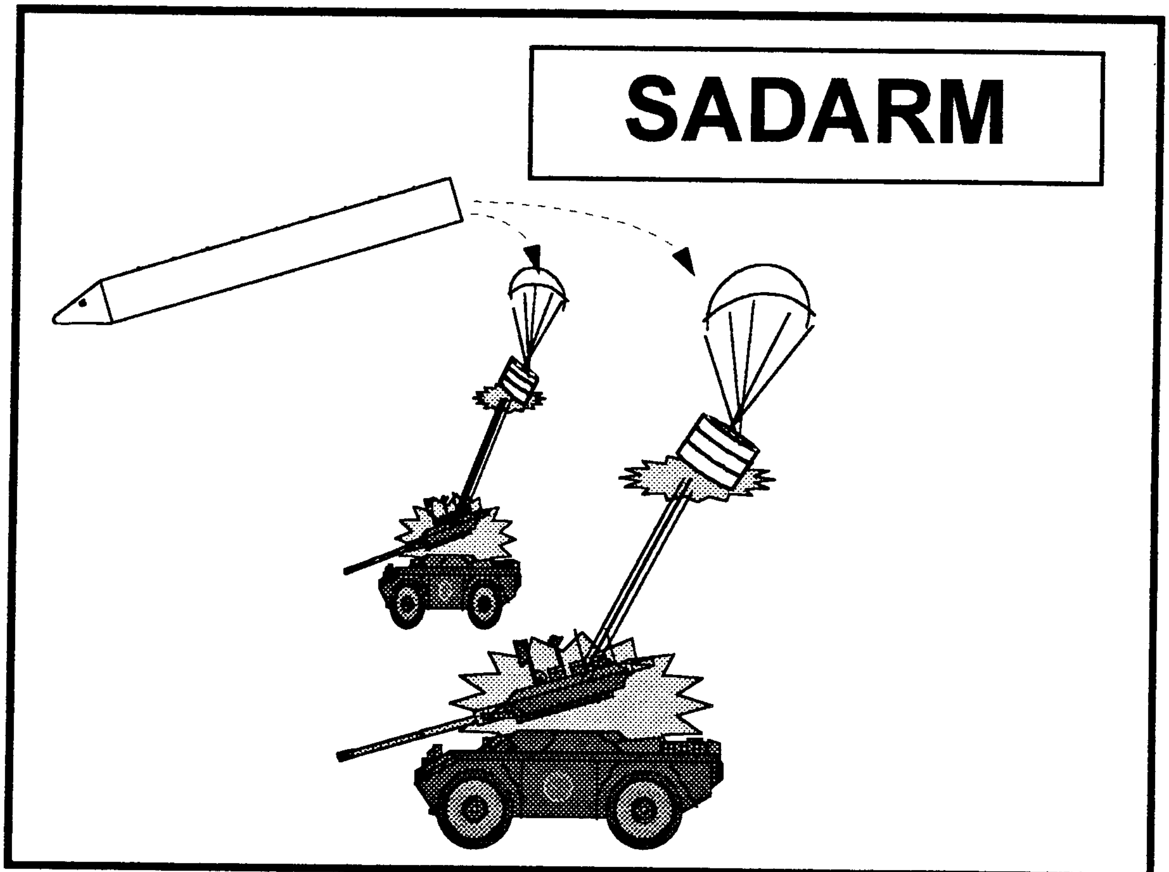

Figure 6. Sense and Destroy Armor (SADARM)

processing selects an armored vehicle to attack. The submunition determines the optimum attack altitude and position. It guides itself over the target. At the selected altitude and

attitude, SADARM fires an explosively formed penetrator into the top of the target. SADARM is capable of defeating all known and projected armor. [Ref. 16]

SADARM is controlled by embedded software [Ref. 16]. The software in SADARM is classified as "firmware" [Ref. 17]. Firmware is defined in DOD-STD-2167A as:

> The combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device. The software cannot be readily modified under program control.

Documentation requirements for firmware are generally less stringent than for embedded software. [Ref. 8]

The software developer for SADARM is Aerojet. SADARM software is embedded inside the submunition. Once SADARM is packed inside the carrier, it becomes a "wooden round." No maintenance of the complete round is required in storage. It is stored and handled as any other common artillery munition. Performance upgrades to SADARM will go into new rounds. Any upgrade to existing SADARM rounds would require remanufacturing. [Ref. 16]

C. SUMMARY

This chapter provided the background information required to understand the thesis questions. It examined the SECDEF's directive to move from military specifications and standards to commercial standards and practices. The development of embedded software was governed by DOD-STD-2167A from February 1988 until December 1994. The majority of software in armament systems was developed with DOD-STD-2167A. On 8 November 1995, MIL-STD-498 replaced DOD-STD-2167A. The major differences between the two standards were also reviewed in this chapter. MIL-STD-498 is only a temporary replacement for DOD-STD-2176A. It will be replaced by a civilian standard in about two years.

Additionally, this chapter reviewed three Army armament programs. The personnel interviewed for the investigation of the thesis questions all participated in the develop-

ment of one or more of these programs. The three programs are Paladin, Crusader and SADARM. The three programs span the acquisition cycle from DEM/VAL to production. Also, the approach to software development was different for each program. Paladin development required the use of military standards. Crusader is using a performance specification, without requiring military standards, to develop its software. SADARM employs firmware. This allows the investigation to look at embedded software development from three distinct perspectives.

III. CONTRACTING FOR SOFTWARE DEVELOPMENT AND MAINTENANCE

This chapter begins the examination of the primary thesis question. What is the effect on armament system acquisitions of allowing commercial practices to be used instead of requiring DoD or military standards in the development and maintenance of embedded software? This chapter will look specifically at the first of the five subsidiary questions. How will this change in policy affect the way PMs and other government mangers contract for software development and maintenance?

The data for this question was conducted by interviews. They were conducted in two phases. During the first phase, the interviewee was provided the thesis questions and given background material on the change in policy from DoD and military standards to commercial practices. The second phase followed one to four weeks later. It consisted of a traditional interview centered on the primary and subsidiary thesis questions. Several of the persons interviewed supplied both written and verbal comments.

Twenty-one individuals were interviewed. All are well-experienced software and acquisition professionals. The average professional interviewed had over 25 years experience in software development and maintenance and over 15 years experience in program and project management. All were familiar with one or more of the three armament programs presented in Chapter II -- Paladin, Crusader and SADARM.

Thirteen of the persons interviewed were government civilian officials. They ranged in grade from GM/GS-14 to members of the Senior Executive Service (SES). All of the government officials participate in the oversight, development, management, and/or evaluation of embedded software acquisition and maintenance contracts and programs. The background of the government professionals is approximately evenly divided among product development (largest concentration), quality assurance, software maintenance and test and evaluation. Most of the professionals had backgrounds that covered at least two of these disciplines.

The remaining eight persons interviewed were from industry. Four managed large armament embedded software development projects. Two managed major armament software maintenance efforts. One was a senior software technician and recognized industry expert on the software development process. The remaining industrial professional managed independent verification and validation (IV&V) for a large aerospace engineering firm. All of the industrial professionals had experience in product development, quality assurance and test and evaluation. Most of their experience was with development and maintenance of large military embedded software systems.

A. RESULT OF INTERVIEWS

The answers to the fist subsidiary question from all government software managers interviewed were almost identical. "We will use performance specifications in the Request for Proposal (RFP) that minimize the use of military standards and specifications" or similar words. Unfortunately, the answer to the first subsidiary question becomes more complicated when viewed in the context of the primary question. Most of those interviewed believed the change from DoD and military standards to commercial practices will have an impact. They expected changes in the government contracting process. They also expected the changes in the contracting process to vary over time. No difference was identified between contracting for development or maintenance of embedded software.

Most of the subjects interviewed believe that the policy change could impact preparation of the RFP and the source selection phases in contracting. The RFP solicits proposals from potential contractors. It describes the product the Government desires and the basis for awarding a contract. Source selection includes analyzing the contractor's proposals and deciding which proposal best meets the Government's needs. It is the process of determining which proposal will be awarded a contract.

The majority of the deputy PMs interviewed identified two concerns with preparation of the RFP. First, there is no civilian equivalent to MIL-STD-498. Second, govern-

ment acquisition professionals and technicians lack experience with civilian standards and specifications.

The government software standards, MIL-STD-498 and its predecessor, DOD-STD-2167A, define the documentation required and alternative processes for development and maintenance of software [Ref. 8]. There is no single civilian equivalent [Ref. 7]. For instance, no civilian industrial standard or specification covers the entire life cycle of software. Existing industrial standards and specifications deal only with individual requirements for documentation or process [Ref. 8]. For example, ISO 9001, Quality System - Model for Quality Assurance in Design/Develop-ment, Production, Installation, and Servicing, and ISO 9003, Guidelines for Application of ISO 9001 to the Development, Supply, and Maintenance of Software, deal only with quality assurance requirements [Ref. 8]. Additionally, ANSI/IEEE Standard 830, Recommended Practice for Software Requirements Specifications, addresses only preparation of the SRS [Ref. 8]. Previously, most RFPs relied on military standards to describe government requirements for embedded software.

Many of the civilian and government software professionals interviewed are concerned, that without an over-arching industrial standard, the Government will be forced to use MIL-STD-498 or risk inadequate specification of requirements. The greatest apprehension expressed by the government professionals is that software developed or maintained without an adequate standard could not be readily maintained in the future.

Most of the civilian and government software managers interviewed also expressed doubt about the familiarity of government technicians with existing civilian standards and specifications. The software managers were concerned that civilian standards and specifications might be inappropriately applied in RFPs. Improper application of standards and specifications can lead to either inadequate specification or over specification of requirements. Neither outcome is desirable.

The software managers were mainly concerned with three issues during source selection.

- What is the impact of dealing with multiple industry standards and specifications?
- Are those standards and specifications adequately applied in contractor's proposals?
- Are government personnel familiar enough with those standards and specifications to evaluate their application?

MIL-STD-498 relates 23 topics/requirements to civilian standardization documents. Seven of these topics/requirements presently have no civilian equivalent standard or specification. Other single topics/requirements link to as many as four different civilian industrial standards and specifications. Only three correspond to a single commercial standard or specification. A partial representation of the information in MIL-STD-498 is provided in Table 1. Thirty different industrial standards and specifications are correlated

Topic/Requirement and MIL-STD-498 Paragraph	**Related Civilian Standardization Documents**
Software Quality Assurance (5.16)	ISO 9001, Quality System - Model for Quality Assurance in Design/Development, Production, Installation, and Servicing ISO 9003, Guidelines for the Application of ISO 9001 to the Development, Supply, and Maintenance of Software ANSI/IEEE Std 730, Standard for Software Quality Assurance Plans IEEE Std 1298/A3563.1, Software Quality Management System
Systems Engineering (5.1.3, 5.3, 5.4, 5.10. 5.11)	None
Software Requirements (5.3.3, 5.5)	ANSI/IEEE Std 830, Recommended Practice for Software Requirements Specification

Table 1. Related Civilian Standardization Documents [After Ref. 8]

with MIL-STD-498. Standards and specifications linked to topics may not be compatible with each other; even though they are compatible with the intent of MIL-STD-498. [Ref. 8]

The new policy implementing the use of commercial standards allows each contractor to select the civilian or military standards they will apply in their proposal. It is probable that no two proposals will employ the same standards. The government software managers interviewed indicated that in a single source selection evaluation they might receive proposals that would require familiarity with all thirty of the referenced industrial standards and specifications. They implied that dealing with multiple standards and speci-

fications complicates source selection. Most of the managers interviewed believe that dealing with multiple standards is more difficult than dealing with a single military standard. Additionally, they indicated that it is more difficult to compare proposals that use different standards to accomplish similar tasks. Those managers believe the increased complication will require additional time and/or qualified personnel to adequately evaluate the application of civilian standards and specifications in proposals. Additional time and/or personnel will increase the cost of source selection. It will also likely increase protests.

Some of the government managers believe that many of their supporting technicians are not familiar with civilian industrial standards and specifications for software development. They feel that this lack of familiarity may lead to incorrect evaluation of contract proposals during source selection. This could result in the Government accepting less than the best proposal.

The IEEE and EIA are working together to produce a commercial equivalent to MIL-STD-498. This IEEE/EIA standard is scheduled for approval within two years. Because of this, many government and civilian software managers stated they believe the impact of changing to civilian practices will change over time. This attitude was best expressed by Mr. Dan Nathan, Chief, Life Cycle Software Support Center, Picatinny Arsenal.

> I believe that the impact of changing to commercial software practices will have no major effect on the way we do business in the long run. We will go through a two to three year transition period while an acceptable civilian standard is developed. After the civilian standard is accepted and we gain experience with it, we will use it in the same manner as we use DOD-STD-2167A... We need to take care in the transition period and influence the development of the civilian standard. We need to make certain the new civilian standard will meet our needs...

Over eighty percent of those interviewed express the belief that the concerns cited for RFP preparation and source selection apply to the transition period. Those concerns can be mitigated by adoption of an adequate civilian replacement for MIL-STD-498.

They believe that the key to an acceptable replacement for MIL-STD-498 is government participation in development of the new commercial standard.

B. DISCUSSION

It is noteworthy that the representatives of PM Crusader showed less concern regarding the impact of using performance specifications (that do not require the use of military standards) for software development than other government personnel interviewed. This is probably due to their recent experience in this area. Contracting for Crusader development,including software, was based upon a performance specification that did not require the use of military standards. The DEM/VAL Crusader contract was awarded to United Defense. United Defense and its subcontractor, Magnavox, proposed using a tailored implementation of DOD-STD-2176A. Both companies plan to transition to MIL-STD-498.

Others interviewed were quick to point out that Crusader is very early in the acquisition cycle and that contract award was prior to the announcement of the policy change. They believe that it is too early to evaluate Crusader's experience. It can better be evaluated once it has entered EMD.

A common thread runs through the concerns expressed about the Crusader experience, preparation of RFPs, and source selection. That thread is uncertainty. Uncertainty about the ability to deal with industrial standards and specifications during the transition period. Uncertainty about how to implement the new policy. Uncertainty about the new commercial standard. Uncertainty and transition often seem to exist at the same time.

Reducing this uncertainty can smooth the transition from military standards to commercial standards. Education about existing commercial standards and specifications can reduce this uncertainty. Educating and training software managers and technicians on these standards and specifications will reduce the apprehension about their use in proposals. Education should greatly reduce concerns about familiarity with industrial standards and specifications.

Government observation or participation in the development of the new commercial standard is essential and will reduce uncertainty. Government experts will accept the new standard more readily if they participate in its development. They become stakeholders in the new standard; this in turn helps insure its success. The transition to the new standard can be further enhanced if progress toward its development is made public knowledge in the software development community. Just as government participation is necessary, active contractor and civilian software professional organization involvement is required. This will also help reduce uncertainty.

Government policy makers can also speed the transition to civilian standards. This can be accomplished by telegraphing implementation guidance in military and professional publications and forums. This knowledge should improve the general comfort level during the transition.

The Government will continue to contract for software development and maintenance during this transition period. Because no single, over-arching, civilian standard exists, government managers have three options in preparing a RFP.

- Obtain a waiver and require the use of MIL-STD-498.
- Tailor and use MIL-STD-498. Allow contractors to substitute existing industrial standards and specifications in place of individual MIL-STD-498 requirements in their proposals.
- Develop RFPs employing performance specifications, and existing industrial standards and specifications that do not reference military standards.

The first option does not support the spirit of the SECDEF's memorandum implementing the transition from military standards to commercial standards. Senior government policy makers interviewed indicate they will not support this option. They cite Crusader as an example of successful software contracting without requiring military standards.

The second option meets the guidance in the SECDEF's memorandum. It allows the use of industrial standards and retains the benefits of using an over-arching standard.

The final option is the most difficult to perform. It requires the development of a RFP that contains the process and documentation guidance contained in MIL-STD-498. Essentially, a tailored version of MIL-STD-498 must be written into the RFP. This increases the risk of inappropriate specification through omission or over specification of requirements. Additionally, this approach may increase the size and complexity of RFPs. Generally, the larger and more complex the RFP, the more difficult and expensive it is for industry to prepare an adequate proposal.

The second option (using MIL-STD-498 and allowing substitution of commercial practices) is the preferred approach. It meets the requirement to transition to commercial practices and avoids the risk in option three.

C. SUMMARY

This chapter addressed the first subsidiary thesis question. How will this policy change, from DoD and military standards to commercial practices, affect the way PMs and other government managers contract for software development and maintenance?

The apparent answer is that over the next two years MIL-STD-498 will be used and substitution of commercial standards for MIL-STD-498 requirements will be allowed. During this period a new commercial standard will be developed. IEEE and EIA are jointly developing the new standard. This will also allow the development of guidelines and training to implement the new policy.

The impact of the change can be reduced by:

- Educating and training software experts and managers on existing industry standards and specifications.
- Government participation in development of the new commercial standard.
- Early and continuous communication of policy implementation guidance.

Most of the problems in transitioning from military standards to commercial standards will come from uncertainty. The keys to reducing uncertainty are education and communication.

IV. MAINTAINABILITY OF SOFTWARE

This chapter looks specifically at the second of the five subsidiary research questions. How will this change in policy impact the maintainability of armament software?

The importance of software maintenance for armament systems is expressed in a statement by Martin McCaffrey of the Naval Postgraduate School.

> Software maintenance has emerged as perhaps the most important critical issue facing the software professional today. Maintenance has become the most costly part of the software life cycle. [Ref. 18]

Software maintenance is also called Post Deployment Software Support (PDSS). Software maintenance for embedded software begins with production of the system containing the software. The Mission Critical Computer Resources Management Guide defines software maintenance for DoD.

> Post Deployment Software Support is the sum of all activities required to ensure that, during the production/deployment phase of a mission critical computer system's life, the implemented and fielded software/system continues to support its original operational mission and subsequent mission modifications and production improvement efforts. [Ref. 5]

Software maintenance consists of three categories of effort. The first category deals with correcting latent defects. The second category deals with adapting software to perform its original function more efficiently. The last category of software maintenance is modifying software to enhance performance or add new capabilities in response to new requirements. [Ref. 5]

Problems with software maintenance can be categorized into five principal areas.

- Software quality.
- Documentation.
- Users.
- Personnel.
- Management. [Ref. 18 and 19]

Software quality problems result from poor design, poor development procedures and/or previous maintenance. Quality problems are often the result of inadequate methodology or discipline in the development process. Some software quality issues include:

- Inadequate software design.
- Inefficient and/or poorly coded software.
- Poor database design and definition.
- Poor data definitions.
- Use of multiple software languages.
- Increasing resource requirements for software maintenance. [Ref. 19 and 20]

Software documentation is the way current software developers and maintainers communicate with future software maintainers. Documentation communicates the what, how and why of software design and code to future maintainers. Software maintenance is dependent upon good documentation. [Ref. 18 and 19]

Software maintainers rely on the user to define requirements for software enhancement. The user often does not adequately define or prioritize requirements. This can result in software that does not perform as the user desires. [Ref. 19]

Quality software maintenance requires experienced, professional, and committed personnel. A frequent error is assigning maintenance to inexperienced software technicians. [Ref. 19]

Software maintenance depends on effective management and review of the development and maintenance process. Standards have to be established and enforced for process, configuration control, coding and documentation. Software managers must insure that current software development and maintenance efforts anticipate and facilitate future maintenance. [Ref. 18, 19 and 20]

A. RESULT OF INTERVIEWS

All of the software professionals interviewed equate maintainability to documentation. The professionals with quality assurance or test and evaluation backgrounds expound that accurate and correctly formatted documentation is required to ensure main-

tainability. Those with software management and maintenance backgrounds emphasized quality of documentation over format.

The subjects interviewed agree on the basic parameters required for software maintenance. These parameters include:

- The ability to trace software requirements to performance standards.
- The ability to trace software requirements through design to code.
- The software development environment.
- Any CASE tools used in development.
- The compilers used to develop code.

The government professionals related that a <u>correct</u> application of MIL-STD-498 or DOD-STD-2167A guarantees transfer of the above parameters. The civilian professionals agreed with the government professionals with two caveats. The Government often blindly uses military standards, such as MIL-STD-498 or DOD-STD-2176A, without tailoring out unnecessary requirements for documentation. Document formats other than those used by the Government can sometimes provide the same information at a lower cost.

A major concern stated by the interviewees is the lack of an over-arching civilian industrial standard equivalent to MIL-STD-498. This same concern was also raised in Chapter III with respect to contracting for software development and maintenance. Most of the government software managers related that they rely heavily on the process and documentation requirements defined in MIL-STD-498 to build-in maintainability.

Another concern raised by the government professionals is the complexity introduced by the myriad of conflicting industrial standards and specifications. The quantity of industrial standards and specifications and their relationship to MIL-STD-498 was discussed in Chapter III. Government software managers confirmed that their software technicians and software maintenance contractors are familiar with maintaining software documented in standard government format.

The government professionals indicated that the impact of changing from military standards to commercial practices will not, for the most part, become evident for several

years. Government maintenance personnel will begin to maintain software documented with industrial standards and specifications at that time.

The government managers stated, that while the industrial standards and specifications may contain the same information as the military standard, the information will generally be formatted differently. Some of the government managers are concerned that unfamiliarity with the industrial formats may lead to misunderstanding or deficiencies in deliverable documentation. This in turn may increase the cost and time required to support fielded software.

The subjects interviewed believe that software maintenance, like contracting, will experience a transition period while the civilian standard is being developed. They assume the transition period is a natural consequence of adjusting to the change from military standards to commercial practices. They believe that the effect of the changes in contracting for software development and maintenance will not become fully evident for several years.

During the transition period for contracting, the government professionals worry that the quality of documentation for software will decline. The subjects interviewed indicate that in June they were directed to convert from military standards and specifications to commercial practices. Since the direction to change policy, little, if any, guidance on the "how" of implementing that change has been forthcoming. This lack of guidance on "how" creates an atmosphere of uncertainty.

The interviews revealed that uncertainty exists in both the government and civilian software maintenance communities. Government software developers indicated this uncertainty may result in inadequate specification during the RFP/contracting process for documentation required to support software maintenance. The software professionals fear that shortcomings in contracting and development today will translate into future maintenance problems.

The government professionals expressed the opinion that the transition period for software maintenance may be longer than for contracting. This is because the software

developed during the transition period will not require maintenance immediately. They related that embedded armament software generally requires periodic maintenance upgrades at 18 to 24 month intervals following system fielding. It will be several years before the affect is known.

B. DISCUSSION

The foundation for some of the government software professional's trepidation is illustrated by recent events in the Paladin program. Paladin's Automatic Fire Control System (AFCS) embedded software has been modified twice since production began in 1990. AFCS software has been previously modified to support changes in the artillery's Tactical Fire Control System (TACFIRE), introduction of the 155 millimeter SADARM munition and addition of an on-board muzzle velocity chronograph (constant round-to-round measurement of muzzle velocity and correction for muzzle velocity variation in ballistic computation). Paladin's AFCS software is currently being modified again. These modifications support new changes to TACFIRE, an upgrade of the main processors in the hardware, and addition of a laser firing device. The laser firing device automates firing of the cannon by replacing the manually fired percussion primer system used today. [Ref. 21] This serves as a perfect example that software maintenance will be significant for the systems of the future.

The AFCS software was developed and documented using DOD-STD-2167A and its predecessors [Ref. 13]. According to both the government and civilian software professionals, the maintenance of AFCS embedded software has been relatively easy and inexpensive considering its size and complexity. It was developed using the Ada language [Ref. 21].

The Modular Azimuth Positioning System (MAPS) performs position location, tracking during aiming, and navigation for Paladin. MAPS is a ring laser gyroscope inertial navigator. MAPS employs embedded software to perform its functions and interface with the AFCS. The amount of software is much smaller than the AFCS software. The modification of MAPS hardware and software is managed by PM Paladin. It is being

modified to incorporate an automatic interface with the Global Positioning System (GPS). In addition to Paladin, several radar and missile systems also use MAPS. [Ref. 21]

MAPS was acquired as a "black box." This means that MAPS was acquired with a form, fit and function specification rather than a detailed development specification. [Ref. 13]

Since MAPS was a "black box" procurement, the Government did not supervise the software development or acquire software documentation. MAPS embedded software was developed using the developer's commercial software practices. MAPS is contractually maintained by its developer and producer, Honeywell. [Ref. 13]

Both military and civilian software professionals indicated that the MAPS software would be much more difficult to maintain than the AFCS software. The principal reason cited for the increased maintenance challenge stems from the lack of adequate documentation for the MAPS software. The commercial standards used during MAPS development permitted much less rigorous documentation than software developed under military standards [Ref. 21].

Paladin's software managers point out the integration of GPS with MAPS is not proceeding as smoothly as the AFCS software upgrade. They stated Honeywell's MAPS development team was disbanded five years ago when the system entered fielding. Honeywell's current MAPS team includes no members of the original team. Paladin's managers believe the lack of thorough documentation has hindered the software update. They stated the MAPS upgrade is less complicated than the Paladin improvements. Both software maintenance efforts began at about the same time and the MAPS upgrade is three to six months behind the Paladin effort.

Military software professionals cited the Paladin experience as evidence why present commercial practices are inadequate or inappropriate for use with military embedded software programs that are to be maintained by the Government. The civilian professionals point out that the government acquired MAPS without intending to maintain its software. Since Honeywell knew the Government had no intention of maintaining MAPS

software, it was not developed and documented to accommodate maintenance or modification requirements. If the military wants maintainable software it must include such requirements in its contract award process and incur the additional cost. The civilian professionals stated you pay for software maintainability: either upfront in development or later in PDSS when the costs will be higher.

The Paladin experience points out the need to educate government software developers on issues to be aware of when commercial practices are used. It may also be used to support the case for involving DoD software maintenance professionals when developing the new civilian industrial software standard. Co-development will reduce the uncertainty about the future, and at the same time force DoD to carefully review its software maintenance development requirements.

Education about existing commercial standards and specifications can reduce uncertainty during the transition period. The probability of the improper or inadequate application of civilian standards and specifications can be significantly reduced if both DoD and civilian software maintainers understand these standards and specifications and identify government maintenance inadequacies.

As stated in Chapter III, MIL-STD-498 can be used with commercial standards during the transition period. This may alleviate many of the concerns about process and documentation raised by the software maintainers.

DoD policy makers can also reduce the turmoil of transition. Turmoil is in part a result of uncertainty about future policy application. This uncertainty can be reduced by early and frequent distribution of information about policy changes during the transition period. Policy change should be documented in DoD and civilian professional publications. If possible, it should be held to a minimum. Information about the change from military standards to civilian standards and how to apply them will reduce turmoil and smooth the transition period.

Several of the government managers cited a recent action by the Army that helped condition them for the change to commercial practices. AMC began several years ago to

review prospective RFPs. The AMC reviews looked for ways to reduce cost in government development programs by eliminating unnecessary government requirements. The review panel encouraged allowing contractors to propose commercial practices in place of military standards. The managers indicated their experience with the this process was positive. None of the managers believed that the panel actions reduced the maintainability of software in the programs reviewed. The AMC panel was led by Mr. Darold Griffin. He recently led the SECDEF's PAT that recommended the change from DoD and military standards to commercial practices. Several systems developed under this initiative are about to enter the software maintenance cycle, so the impact should be known soon.

Positive experience with commercial practices in military software development programs needs to be consolidated and communicated in DoD and civilian technical publications. The communication of positive and negative development and maintenance experience with commercial practices can help smooth transition from military standards to civilian practices.

C. SUMMARY

This chapter addressed the second subsidiary thesis question. How will this change in policy, from DoD and military standards to commercial practices, impact the maintainability of armament software? .

There will not be immediate impact on maintainability during the two year transition period. Software developed and/or maintained with the new commercial standard is several years away from entering the maintenance cycle.

The impact of the change can be reduced by:

- Emphasizing the requirement for quality of documentation over format.
- Educating software maintenance professionals and managers on existing industry standards and specifications.
- Government software maintainer participation in development of the new commercial standard.
- Bridging the transition period with MIL-STD-498.
- Communication of early successes with software maintenance.

- Early and continuous communication of policy implementation guidance.

The keys to success for software maintenance are well educated and quality personnel and managers, and the production of high quality, well documented, embedded software for armament systems. Software must be developed, documented and maintained to meet future potential maintenance requirements. This is especially true during the transition period.

V. TEST AND EVALUATION OF SOFTWARE

This chapter looks specifically at the third of the five subsidiary research questions. How will this change in policy influence the test and evaluation of armament software?

Software test and evaluation are inherent in the software development and maintenance process defined by MIL-STD-498 and its predecessor DOD-STD-2167A [Ref. 3 and 8]. Software test and evaluation consist of:

- Developmental Test and Evaluation (DT&E).
- Operational Test and Evaluation (OT&E).
- Formal process reviews and audits [Ref. 5 and 8].

Many software texts and articles address only the DT&E aspects of testing. However, the software professionals interviewed generally expressed the view that DT&E, OT&E and formal reviews and audits are all important components of software test and evaluation.

Embedded software DT&E is an independent component of the overall system's DT&E. Software DT&E is conducted during each phase of software development. It uses both a top-down and bottom-up approach to test and evaluation. A simplified view of software test and evaluation is illustrated in Figure 7. [Ref. 3 and 5]

During the systems requirements phase of development, system requirements are allocated to software at the top level. Software test planning begins. Software design divides software requirements into smaller requirements. Test plans become more detailed. This process continues as these requirements are divided into smaller requirements. The test plans become even more detailed. When the individual requirements are reduced to executable software elements (single small tasks), they are implemented as code. Walk-throughs, audits and reviews are conducted to evaluate the accuracy and adequacy of software design, test plans and code throughout the design process and into coding. Code testing begins with the lowest elements. The software units are assembled into larger units (units, modules and computer software configuration items) and tested.

This assembly from the bottom-up and testing continues until the assembled software program is tested. [Ref. 5]

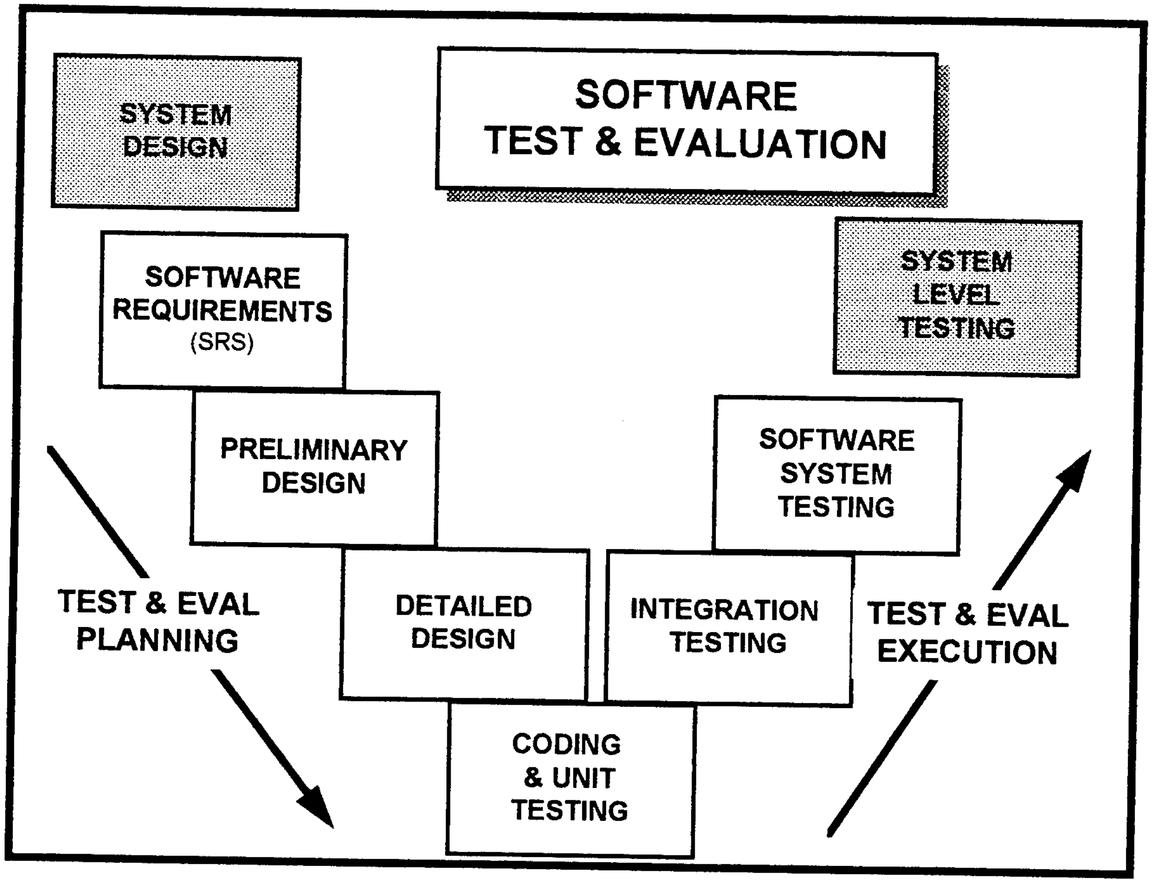

Figure 7. Software Developmental Test and Evaluation

During DT&E, software is usually initially tested separate from the hardware. Following software testing the software is combined with its target hardware and integration testing is conducted. This process helps isolate hardware errors from software faults. It simplifies fault correction. A major goal during DT&E is to detect faults so they can be corrected before the system enters production. The earlier a fault is detected, the easier it is to fix. DT&E evaluates software's ability to meet technical requirements. [Ref. 5]

Software OT&E is conducted during the system OT&E. During OT&E, embedded software is evaluated on its capability to perform as a component of the overall system. The goal of OT&E is to determine if the overall system, including software, meets

requirements when operated by users in an operational environment. Inadequate or unreliable embedded software results in an unsatisfactory system. [Ref. 5]

Software test and evaluation documentation is produced as a part of the overall documentation for software. Each phase of development terminates with a formal review or audit. Figure 2, Chapter II illustrates the relationship among the development phase, documentation and formal reviews and audits. The review or audit evaluates readiness to begin the next phase in development. These reviews and audits are an important part of embedded software test and evaluation. [Ref. 8]

Software testing during each development phase is usually initiated by the contractor. Final software testing in a phase is usually conducted by the Government. In some cases software testing may be conducted by a contractor/government team during DT&E. OT&E must be conducted by the Government. [Ref. 5]

A. RESULT OF INTERVIEWS

The viewpoints of the software professionals interviewed split based upon the test experience of the individuals. Several of the government professionals had experience in planning and conducting OT&E. The majority of the government and civilian professionals interviewed were experienced with DT&E. All had experience in formal reviews and audits.

The government professionals with OT&E backgrounds stated the key software development product for them was the SRS. They indicated the SRS provides them with the link between system requirements and performance standards and software requirements and performance standards. These professionals stated they use the SRS with system level requirement documents to plan and conduct OT&E. The software professionals with OT&E backgrounds stated they expect little impact from the policy change to commercial practices as long as an equivalent to the SRS exists.

The remaining software professionals have a very different opinion about changing from military standards to commercial standards and specifications. They related that they

relied heavily upon DOD-STD-2167A and expected to rely on MIL-STD-498 to insure that requirements and performance standards are traceable and documented from the overall system level to the code. These professionals stated this traceability is required to plan and conduct an effective test and evaluation program for embedded software.

The software professionals with DT&E experience stated that effective embedded software testing measures the performance of the code and compares that performance with the performance standards in documentation. One of the civilian professionals noted "while you can test software at the system level -- it is very difficult to isolate and correct faults at the system level." The DT&E professionals interviewed related that it is much more effective to test embedded software in small units rather than the whole. In order to test at the small unit level, they stated, you needed to know the designed performance at that level.

As discussed earlier in Chapters III and IV, the software professionals pointed out there is no civilian standard or specification equivalent to MIL-STD-498. According to the subjects interviewed, existing commercial software test standards can substitute effectively for most of the individual MIL-STD-498 test requirements. However, they relate that no existing commercial standard or group of commercial standards can adequately replace MIL-STD-498 for requirement traceability.

Those interviewed are aware that IEEE and EIA are developing a commercial standard to replace MIL-STD-498. They indicated that an adequate commercial standard could be developed. However, government experts with DT&E backgrounds interviewed expressed apprehension that the new commercial standard might not adequately address requirement traceability.

As expressed in Chapters III and IV, nearly everyone interviewed believes that a transition period will exist until the new commercial standard completes development and is enacted. They further indicate that test and evaluation problems may increase above normal during the transition period. The major reason for the potential increase in problems cited was "how" to ensure requirement traceability during transition. This is when

commercial standards are applied during the requirements analysis and detailed design phases of development.

B. DISCUSSION

The interviews indicate very little concern from operational testers about the change to commercial standards. The major issue raised by operational testers was the traceability of system requirements to the software. The SRS is the document most often used to link system requirements and performance standards to software. Testers use this information to select and design test instrumentation, develop detailed test issues and criteria, plan testing, and assign responsibility for test incidents (shortcomings and failures). They stated ANSI/IEEE Standard 830, Recommended Practice for Software Requirements Specifications, produces acceptable documentation. This commercial specification replaces the DOD-STD-2167A or MIL-STD-498 DID for the SRS. However, they noted that no commercial standard or specification addresses the requirements analysis process.

Four commercial standards address software testing. These standards address the development of test plans and execution of software developmental tests from unit through system software testing. The standards are:

- ANSI/IEEE Standard 829, Standard for Software Test Documentation.
- ANSI/IEEE Standard 1008, Standard for Software Unit Testing.
- ANSI/IEEE Standard 1012, Standard for Software Verification.
- IEEE Standard 1059, Guide for Verification and Validation Plans. [Ref. 8]

Most of the government and all of the civilian software professionals interviewed were familiar with these commercial standards. They expressed the opinion that these standards were acceptable if applied properly. Several of the government developmental testers stated they observed and participated in the successful use of these standards during informal software testing for Paladin. Much of Paladin's developmental software testing was conducted in the contractor's facility by a government/contractor test team.

As with operational testers, government developmental testers were concerned about requirement traceability. The major issue about transitioning to commercial

standards was that requirements analysis is not presently addressed by a commercial standard. The government testers indicated that using MIL-STD-498 and replacing test requirements, where applicable, with commercial standards could result in an adequate software test program.

The government professionals interviewed felt that government participation in development of the commercial standard to replace MIL-STD-498 was desirable. They noted that software testing is expensive. They were concerned that, in an attempt to control software cost, software managers might reduce testing requirements below acceptable levels in the new standard. They felt there is a correlation between software quality and the quality of testing conducted.

From the interviews it appears the government test and evaluation community is the least affected by the change to commercial standards. They raised fewer issues than any other group. They were also the only government group interviewed who indicated they were familiar with all of the existing commercial standards referenced by MIL-STD-498 for their area of expertise. They felt that if the other phases of software development were properly executed and documented, they could perform their function with the existing commercial standards during the transition to commercial practices.

C. SUMMARY

This chapter addressed the third subsidiary thesis question. How will this change in policy, from DoD and military standards to commercial practices, influence the test and evaluation of armament software?

Test and evaluation may be affected by changes in contracting for embedded armament system software. Until the new commercial standard is approved, the Government should use MIL-STD-498 and allow substitution of properly tailored commercial standards for test and evaluation requirements.

From a testing perspective, the impact of the change from military standards to commercial standards can be reduced during transition by:

- Emphasizing the importance of requirement traceability in software development during the transition period.
- Ensuring Government test community representation in development of the new commercial standard.

Test and evaluation of embedded armament system software are necessary to ensure the system, with its software, will perform its intended purpose and meet the user's requirements. Tracing requirements and performance standards from the system level through the software documentation to the code is necessary to ensure a sound embedded armament system software test and evaluation program. This capability must be preserved during the transition from military standards to commercial practices.

VI. SOFTWARE CONTRACTORS

This chapter looks specifically at the fourth of the five subsidiary research questions. How will this change in policy affect potential government contractors for armament system software?

Contractors are a very important part of embedded armament software acquisition. All of the software supporting initial production for Paladin, Crusader and SADARM was or is currently being developed by contractors. Of the software for these three systems, the Government developed "in-house" only the modification to support laser ignition for Paladin. Contractors develop and maintain virtually all embedded armament system software. [Ref. 1, 14, 16 and 22]

A single development effort may involve a number of software contractors. Paladin involved seven different software contractors with significant responsibility during development. They are listed in Table 2. Most of these software contractors and the Paladin "Prime" contractor, BMY, employed other software contractors as consultants or subcontractors for lesser software activities contained within their efforts. [Ref. 1]

Contractor	Activity
Alliant Techsystems	AFCS Developer -- System Software Integration -- Software Maintenance
Honeywell	MAPS Developer -- Software Maintenance
GE	Embedded Prognostics/Diagnostics -- Automated Test Equipment
ECC	Embedded Training -- Training Devices
Teledyne Brown Engineering (TBE)	Independent Validation and Verification (IV&V)
Advanced Science Technologies (AST)	IV&V -- Interface Specifications
TELOS	Interface Specifications -- Modifications to BCS and TACFIRE -- User Interface Requirement (UIR)

Table 2. Paladin Software Contractors

As demonstrated in the Paladin example, software contractors are involved in embedded armament software from the initial user requirement (UIR), through

development, and into software maintenance. They are involved throughout the entire life cycle of software for armament systems.

As mentioned earlier in Chapters II and III, the policy change requires the DoD to change the way it contracts for products and services including software. Two of the goals are to open military software development to potential contractors who do not traditionally compete for military contracts, and to reduce cost.

The contractors interviewed are all currently involved in executing contracts for embedded armament system software. Some of the contractors develop and maintain software in both the civilian and military sectors. Others do all, or nearly all, of their software business in the military sector. Several large, medium and small civilian sector software developers were contacted for interviews. All refused an official interview.

The reason given by most the civilian sector contractors for refusing to participate was that, until the regulations governing DoD contracting are changed, they cannot evaluate their potential for participation in DoD projects. Several stated they have no interest in participating in DoD projects. All refused to elaborate further.

A. RESULT OF INTERVIEWS

The software contractors interviewed indicated they have three major concerns with the policy change to commercial standards and specifications.

- They believe that during the transition to the commercial standards there may be a penalty associated with proposing the use of existing commercial standards.
- They believe the government may have an unrealistic view of cost reduction from changing to commercial standards in software development.
- They are also concerned that the military may try to dominate the standardization process for the new commercial software standard to replace MIL-STD-498.

As discussed in Chapters III and IV, the contractors interviewed indicated they believe some government software technicians are unfamiliar with industrial software standards. They stated government technicians work primarily with military standards and

not industrial standards. The contractors noted that software technicians are often employed in source selection to evaluate proposals. They suggested that because of unfamiliarity with industrial standards on the part of some government technicians, proposals based upon military standards may be more likely to be awarded contracts. This would have the same result as penalizing the use of industrial standards. They believe this behavior is temporary and will be corrected as government technicians gain experience with commercial software standards.

Over half of the contractors interviewed stated the government may have false expectations about cost reductions from changing to commercial software practices. It was pointed out that the costs associated with DoD software are based upon the effort required. They stated that the effort required is not just a function of military or industrial standards. Cost and effort are also a function of reliability, design process and software documentation requirements. Non-software items, such as government accounting procedures and special contractual relationships, such as the Government's ability to unilaterally change contracts, add to the cost and effort of doing business with the military.

The final concern was expressed by only a few of the civilian software professionals. They pointed out their previous experience with the DoD indicates the DoD will attempt to dominate most relationships, negotiations or discussions it has with industry. They cited their experiences in contract negotiations and program reviews as evidence. They indicated that attempts by DoD to dominate discussions could draw out the development of the civilian software standard and may result in resistance to its acceptance by the civilian software community. Unless the new standard is accepted by the civilian software community, it will not achieve DoD's goal of attracting new contractors to military projects.

B. DISCUSSION

The contractor's rationale behind their first concern, that there may be a penalty associated if commercial standards are proposed, was covered in Chapters III and IV.

The contractors related that they were concerned about the apparent unfamiliarity of DoD professionals and technicians with commercial standards and specifications. Some of the contractors implied this could result in prejudicial behavior by DoD personnel during source selection.

Discussions in Chapters III and IV indicated that education should alleviate this contractor concern. DoD needs to ensure government software professionals are familiar with commercial software standards. This can best be accomplished through education and training.

During interviews with contractors on cost, the term "silver bullet" often came up in conversation. Several contractors felt the Government was expecting the change to commercial software practices to suddenly reduce the cost of embedded software. As previously mentioned, they believe the entire framework of the acquisition process must be changed to effect real changes in cost.

Other contractors believe that it is too early to form an opinion on the impact of changing from military standards to commercial practices. They point out that the 29 June SECDEF memorandum also directs changing DoD regulations and instructions to enact the policy change. These contractors mentioned the real impact on cost will come from changes to the DoD Supplement to the Federal Acquisition Regulation (DFARS) and the DoD 5000 series regulations and instructions. The DFARS and DoD 5000 series will determine "how" the policy change is accomplished. These contractors are presently uncertain about the impact of the new government policy.

This uncertainty about government policy implementation can be addressed, as mentioned in earlier chapters, by communication. The DoD needs to address the revision of the DFARS and DoD 5000 series regulations and instructions that enact the transition to commercial standards in professional software and management publications. This early communication will help address the first two contractor concerns. If the software contractor community understands the coming changes, it will have a basis to evaluate

DoD's commitment to the change and the ability of the change to reduce embedded armament software cost.

Based on the interviews with software contractors, current DoD software contractors will continue to participate in current and future software developments. The indication from non-government contractors is they will independently act based upon their assessment of the guidelines implementing the policy as the revised versions of the DFARS and DoD 5000 series are made public. The non-government contractors indicated they rely on professional software and management publications to keep them advised of changes in the software development community.

The contractor's final concern can really only be addressed by government actions during the development and approval of the new commercial standard to replace MIL-STD-498. The DoD needs to carefully select its representatives to the IEEE/EIA meetings drafting the new standard. Interviews with civilian professionals involved in developing the new standard revealed the IEEE and EIA are using MIL-STD-498 as the basis for the new standard. MIL-STD-498 should provide a common ground for DoD and industry cooperation in developing the new standard. Additionally, the DoD should continue to be sensitive to industry during the Government approval process.

C. SUMMARY

This chapter addressed the fourth subsidiary thesis question. How will this change in policy, from DoD and military standards to commercial practices, affect potential government contractors for armament system software?

Most current and potential government software contractors indicated they will take a "wait-and-see" position on the policy change. Current government contractors will continue to work on DoD software projects. Prospective government contractors will react based upon their assessment of DoD actions implementing the policy change.

The impact of the change from military standards to commercial standards can be reduced by the following actions:

- Educate and train government software professionals and technicians on existing industrial standards for software development and maintenance.
- Provide early and continuous communication of implementation guidance in professional software publications.
- Ensure DoD and commercial software developers equally participate in development of the new standard.

Contractors are an essential part of the embedded armament software community. Communication is critical to maintaining good government/contractor relations.

VII. RISK AND THE POLICY CHANGE

This chapter covers the last of the five subsidiary research questions. Will the change in policy affect risk in the development and maintenance of armament systems?

The Defense Systems Management College (DSMC) defines risk in Risk Management Concepts and Guidance as:

> Risk is the probability of an undesirable event occurring and the significance of the consequence of the occurrence. This is different from uncertainty which considers only the likelihood of occurrence of the event. [Ref. 23]

Capers Jones, Chairman of Software Productivity Research Incorporated, and a well-known international software consultant, defines software risk in his book, Assessment and Control of Software Risks.

> This term (software risk) means the probability that a software project will experience undesirable events, such as schedule delays, cost overruns, or outright cancellation. Risk is proportional to size and inversely proportional to skill and technology levels. In considering aspects of risk for large systems, the risk of schedule slippage approaches 100 percent, since most such systems are late. The risk of cost overruns is greater than 50 percent. The risk of outright failure and cancellation is about ten percent, since one out of every ten large systems begun in the United States will not be finished and delivered. [Ref. 24]

From the two definitions one may conclude that software risk consists of two components -- an undesirable event and the probability of that undesirable event occurring. The level (quantity) of risk is based upon evaluating the severity of an undesirable event in the light of the probability of its occurrence. Figure 8 illustrates this concept. [Ref. 23 and 24]

The level of risk is often expressed as low, medium (moderate) or high. For example, an event that has a low likelihood of occurrence and an insignificant consequence is clearly a low risk. Likewise, an event with a high probability of occurrence and a catastrophic consequence is a high risk. Events between these two will range from low to high. Determining if risk for a specific event is low, medium or high often depends upon

the perspective of the manager evaluating risk. For example, the manager of a firmware project may consider the use of complex code a moderate risk since firmware often does not require maintenance. However, using that same code in the embedded software for an armament system, that will be modified every 18 to 24 months, may be considered a high risk. [Ref. 23 and 24]

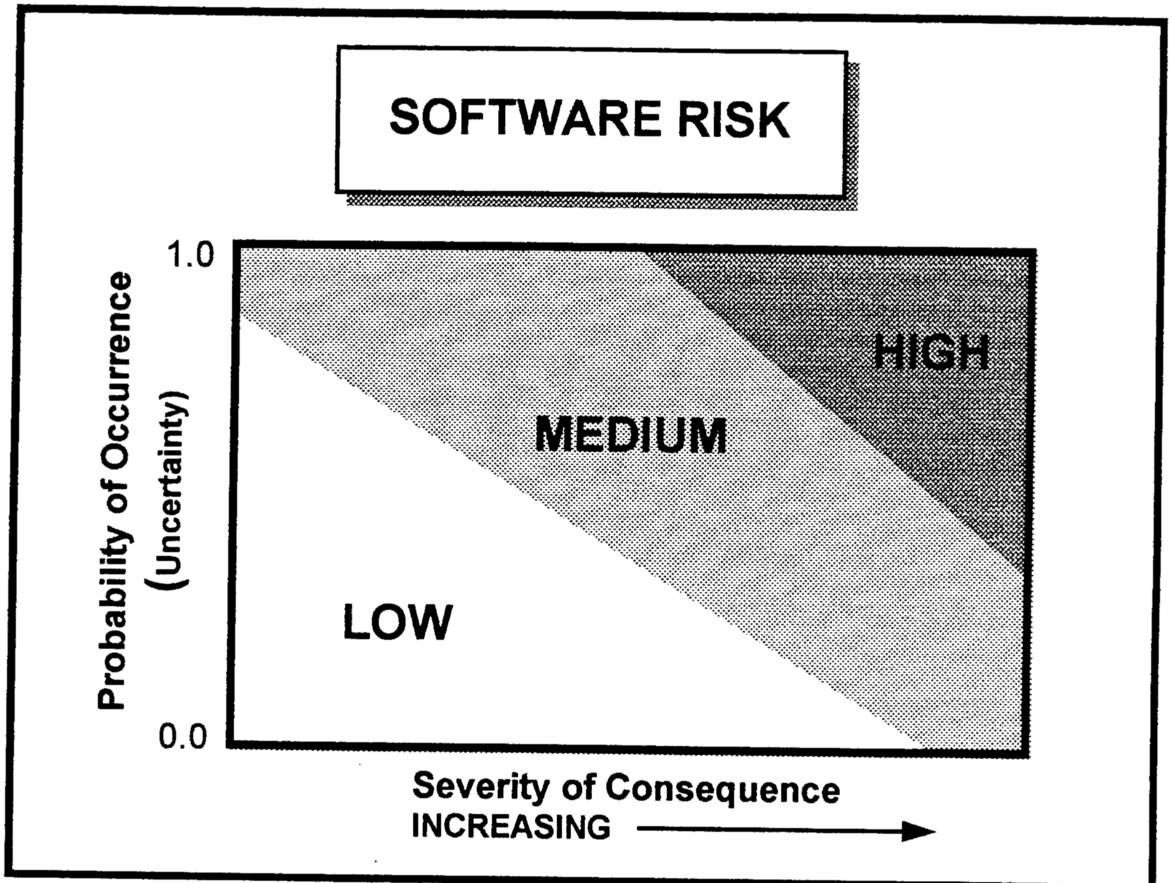

Figure 8. The Level of Software Risk [after Ref. 23]

Software risk management is the set of actions performed to identify potential risks and to eliminate or mitigate their effects. Risk management consists of two phases, risk assessment and risk control. [Ref. 25]

Risk assessment is the study of potential undesirable events (risk factors) in order to determine the probability and significance of their occurrence. Risk assessment requires a software manager to determine the shade of gray, i.e., the manager must determine if a

risk factor presents a low, medium, or high risk and its relationship relative to other risks of the same level. [Ref. 23 and 25]

To assess risk, a software manager must know or be able to estimate the consequences of an event and the probability of the event occurring. If a manager cannot determine or make a reasonable assumption about the probability distribution of an event occurring, the manager cannot assess the level of risk associated with the event. Such an event is classified as a potential risk. [Ref. 23 and 25]

DoD classifies risk in order to facilitate the management of cost, schedule and performance goals. Risk assessment during software development and maintenance is based on the ability of the software to perform its required function, be available at a specific time, and be below a specified cost ceiling. Risk assessment is used to support programmatic decisions. If a software project cannot meet its cost, schedule or performance goals, it is often canceled. Risk assessment is the basis for risk control. [Ref. 23]

Risk control involves both contingency planning and action. It consists of the actions taken or planned to reduce the effects of risk factors before or after they occur. Some risks can be mitigated by preemptive actions. For example, employing design to cost to avoid cost growth. Other risks are mitigated by performing planned actions after the event occurs. As an example, changing to an alternative design option after a less costly design fails to meet a run-time requirement. [Ref. 25]

DoD divides risk into five facets to assist in managing cost, schedule and performance issues. They are:

- Technical -- performance related.
- Supportability -- performance related.
- Programmatic -- environment related.
- Cost.
- Schedule. [Ref. 23]

Technical risk is associated with the ability of software to meet its performance standards [Ref. 23]. It is often software design related. Some examples of armament system software technical risks are the design of algorithms to support meteorological

corrections for ballistic calculations, compilation of a rule base to advise howitzer crews on selection of firing positions, or design of weapon control software to direct weapon pointing in real time.

Supportability risk is associated with the ability to maintain software during and after production [Ref. 23]. Software supportability issues include limiting the complexity of code, planning for PDSS in design, or rigorous documentation of requirement traceability.

Programmatic risk is associated with obtaining the resources necessary to perform and complete a software project [Ref. 23]. Programmatic risks for armament system software projects include obtaining qualified software programmers and managers, obtaining funding, and changing world events (fall of the Berlin Wall and subsequent reduction in defense budget). Sources of programmatic risk can include DoD and service senior management, the congress, the user community, and foreign nations.

Cost and schedule risk are associated with meeting dollar and time constraints. Almost all the other previously discussed risks also impact cost and schedule. The solutions to most technical, supportability or programmatic problems or risks normally involve additional time and funding. [Ref. 23]

Jones takes a different, but not necessarily conflicting view from DoD, in categorizing risk. He divides software risk into sixty risk factors. His risk factors can be associated with one or more of the DoD's five facets. It is interesting to note that he associates one factor, excessive paperwork, solely to military and very large commercial software projects. He states:

> Unfortunately, international standards are a root cause of excessive paperwork. Both DOD-STD-2167A and ISO Standard 9000 - 9004 tend to cause excessive volumes of "boilerplate" or text that is required but not utilized... Software developed with military and international standards often have in excess of 400 words in documentation for every line of code. [Ref. 24]

While Jones attributes DOD-STD-2167A with causing excessive paperwork, he interestingly does support the use of this military standard as a risk reduction measure for

over one third of the sixty risk factors he cites. He suggests it is sometimes necessary to trade one risk factor for another in order to lower the overall risk. He notes that software produced with military standards is usually of high quality and easier to maintain than other software of comparable size and complexity. Additionally, he acknowledges the trend in DoD to reduce unnecessary paperwork by such measures as allowing electronic documentation and tailoring out redundant requirements. [Ref. 24]

Software risk has been briefly discussed in this section. The following section will relate the views of software professionals on the issue.

A. RESULT OF INTERVIEWS

The military and civilian software professionals interviewed stated there is insufficient data available at this time to adequately assess the risks associated with changing from DoD and military standards to commercial practices.

Many of the military software professionals believe supportability, cost and schedule risks may potentially increase during the transition to the new commercial software standard.

The military professionals' rationale for increased supportability risk was addressed in Chapter IV. They indicated that without the discipline and structure imposed by an over-arching standard, such as MIL-STD-498, the quality of software and documentation may decline. They related the belief that this could eventually result in software maintenance problems. It was also noted that potential software maintenance problems represent potential supportability risk. The military professionals pointed out that dealing with problems will increase the time and expense of software maintenance.

The military software professionals also pointed out that any problems that occur in contracting or test and evaluation will probably require additional time and funding to correct. Their concerns about contracting and test and evaluation were covered in Chapters III and V. Concerns related to the use of commercial standards in these two areas included:

- Inadequate specification or omission of requirements.
- Extended time for source selection.
- Loss of traceability for requirements to code.

They noted that each of these concerns has the potential to increase cost and schedule risk.

The commercial software professionals stated that government error and policy change during the transition to commercial standards presented the greatest potential for increased risk. They noted that government omission or errors in RFP preparation and/or changes in policy during the contracting process or after contract award may result in contract modification. They indicated contract modifications usually delay work in progress and change the scope of the contract. They noted scope changes usually increase the amount of work to be done under the contract and therefore increase both the cost and schedule of the software project.

B. DISCUSSION

The change from DoD and military standards to commercial practices was directed in part to mitigate risk. As mentioned in Chapter II, the potential risks addressed by the SECDEF's 29 June memorandum were:

- Decreased DoD budget and increasing cost of military hardware and software.
- Access to commercial state-of-the-art technology.
- Decreasing defense business base. [Ref. 2]

As noted earlier, it is sometimes necessary to trade one risk for another risk in order to lower total overall risk. In the case of the SECDEF's memorandum, mitigation of the three risks above requires assumption of the risk associated with the policy change.

The SECDEF's memorandum contains guidance to mitigate some of the potential risk associated with the directed policy change. The risk reduction measures contained in the memorandum included:

- Authorizing waivers for on-going solicitations and contract negotiations.
- Authorizing waivers when no existing non-government standard or specification exits or use of a government standard or specification makes economic sense.

- Directing rapid changes to existing regulations and instructions to implement the change.
- Encouraging contractors to recommend use of existing commercial practices.
- Encouraging DoD and industry partnerships to develop replacements for military standards and specifications.
- Requiring revision of DoD education and training programs to incorporate specifications and standards reform.
- Directing the programming/reprogramming of funds to implement the change. [Ref. 2]

Temporary approval of MIL-STD-498 and government support for the development of the IEEE/EIA industrial standard to replace MIL-STD-498 represent additional risk mitigation measures enacted since the SECDEF's memorandum.

Each of the concerns or issues stated by the government and civilian software professionals in Chapters III through VI represents a potential risk resulting from the policy change. The answers to the first through forth research questions and the actions recommended to reduce their impact covered in Chapters III through VI represent potential risk mitigation measures for these issues and concerns. A summary of the major concerns and answers is in Table 3.

Concerns (Potential Risk)	**Answers** (Risk Mitigation Measures)
Government unfamiliarity with commercial standards.	Educate and train government software professionals on tailoring and use of commercial standards.
No over-arching civilian software standard.	Use MIL-STD-498 until a commercial equivalent is developed. Tailor and substitute existing commercial standards for MIL-STD-498 requirements where appropriate.
Unknown implementation guidance.	Early and continuous communication in government and professional journals and forums of implementation guidance.
Inadequate or inappropriate new commercial software standard.	Government participation in IEEE/EIA development of the new standard to replace MIL-STD-498.

Table 3. Summary of Major Concerns and Risks

Major General Irby, PEO Aviation, during a recent presentation at the Naval Postgraduate School said, "There are three kinds of risk; known risks, unknown risks and unknown unknowns" [Ref. 26]. Known risks are those with known probabilities of occurrence and consequences. Unknown risks are potential risks. The event has been identified, but the probability of occurrence and/or consequence is not fully defined. Unknown unknowns are potential risks that have not been identified.

Based on the interviews and the above definitions, there are no known risks resulting from the policy change at this time. While concerns have been identified, the probabilities of occurrence and/or consequences are not now known. However, as mentioned above, there are a number of unknown or potential risks. These issues and concerns will remain potential risks until their probability of occurrence can be determined. At this time, there is insufficient experience with the new policy to determine their probability of occurrence.

These potential risks can be addressed by the actions recommended in the previous chapters. These actions are summarized in Section C.

At least one potential risk was not mentioned by the military and civilian software professionals interviewed. The SECDEF's memorandum and the approval memorandum for MIL-STD-498 established a two year limit for publishing changes to DoD regulations and instructions required to fully implement the policy change, and approval of a new commercial software standard to replace MIL-STD-498 [Ref. 2 and 4]. The potential risk is that at the end of the time period either the changes to DoD regulations and instructions or the new commercial standard will not be approved. This would require DoD to extend the transition period until the actions are complete.

Software professionals and managers need to be aware of the fact that unknown unknowns exist. As they gain experience with the new policy, unknown unknowns will, from time to time, materialize and become unknown risks. As a result, risk management is required throughout the development and maintenance cycles of software.

C. SUMMARY

This chapter addressed the last subsidiary thesis question. Will this change in policy, from DoD and military standards to commercial practices, affect risk in the development and maintenance of armament systems?

The answer is certainly yes. However, the extent of the change in risk cannot be determined at this time. Additional experience with the new policy is required to gather sufficient data so that potential risk factors can be identified and their consequences and probabilities of occurrence evaluated.

The impact of the potential risks resulting from the transition to commercial standards can be reduced by the following actions:

- Tailor and use MIL-STD-498,
- Encourage substitution of properly tailored commercial standards for individual MIL-STD-498 requirements, where applicable, while the commercial software standard is being developed.
- Educate and train government software professionals on the tailoring and use of existing commercial standards and specifications.
- Active participation by government representatives in the effort by IEEE/EIA to develop a commercial software standard to replace MIL-STD-498.
- Early and continuous communication in government and civilian software and acquisition professional forums about lessons learned on the policy change and policy implementation guidance.
- A concerted effort to ensure that the commercial software standard replacing MIL-STD-498 is completed and approved within the two year transition period.
- Completion of the necessary changes to the DFARS and other DoD regulations and instructions within the specified transition period.

While risk will undoubtedly result from the change to commercial software practices, the effect of such risk can be mitigated through actions now. Government and industry should form a partnership to identify and control potential risks that arise in software development and maintenance.

VIII. CONCLUSIONS

The conclusions from this study of the impact of adopting commercial practices in software development and maintenance on Army embedded armament systems are presented in this chapter. The chapter consolidates the answers to the primary and subsidiary research questions. It also makes recommendations to reduce the impact of the change. Lastly, recommendations for further research are made.

A. ANSWERS TO RESEARCH QUESTIONS

The primary research question was addressed through investigating five subsidiary research questions. The answers to these research questions and recommendations for each question are presented below.

1. First Subsidiary Question

How will the policy change affect the way PMs and other government managers contract for software development and maintenance?

The answer is that over the next two years MIL-STD-498 will be used and, when appropriate, commercial standards for MIL-STD-498 requirements will be substituted. During this period a new comprehensive commercial software standard will be developed. IEEE and EIA, with government participation invited, are jointly developing the new standard. Provisions for DoD to develop guidelines and training to implement the new standard are planned.

The following actions are recommended to reduce the impact of the change from military standards to commercial standards:

- Educate and train software experts and managers on existing industry commercial standards and specifications.
- Ensure government participation in development of the new commercial software standard.
- Provide early and continuous communication in military and professional publications and forums of policy implementation guidance.

Most of the problems in contracting for embedded armament software during the transition from military standards to commercial standards will result from uncertainty. The keys to reducing uncertainty are education and communication.

2. Second Subsidiary Question

How will this change in policy impact the maintainability of armament software? .

The impact of the change will not be immediate, but occur during a two year transition period and beyond. The full impact will not be known until software developed under the new commercial standard policy has been delivered and has to be maintained.

This does not mean nothing should be done in the interim. The following actions are recommended to reduce the impact of the change from military standards to commercial standards:

- Emphasize the requirement for quality over format in software documentation.
- Educate and train software maintenance professionals and managers on existing industry standards and specifications.
- Ensure government software maintainer participation in development of the new commercial software standard.
- Bridge the transition period with MIL-STD-498. Tailor and use MIL-STD-498 and encourage appropriate substitution of tailored commercial software standards for requirements in the military standard.
- Communicate early successes with commercial standards in military software maintenance in government and professional software publications and forums.
- Provide early and continuous communication in government and civilian professional journals of policy implementation guidance.

The keys to success in software maintenance are well educated and quality personnel and managers and the production of high quality, well documented, embedded software for armament systems. Software must be developed, documented and maintained to meet future maintenance requirements. This is specially true during the transition period.

3. Third Subsidiary Question

How will this change in policy influence the test and evaluation of armament software?

Test and evaluation may be affected by changes in contracting for embedded armament system software. Until the new commercial standard is approved the Government should tailor and use MIL-STD-498 and allow substitution of properly tailored commercial standards for test and evaluation requirements.

The following actions are recommended to reduce the impact of the change:

- Emphasize the importance of requirement traceability in software development during the transition period.
- Ensure Government test community representation in development of the new commercial standard.

Test and evaluation of embedded armament system software are necessary to ensure the software will perform its intended purpose, is properly interfaced with system hardware, and meets the user's requirements. Tracing requirements and performance standards from the system level through the software documentation to the software code is necessary to ensure a sound embedded armament system software test and evaluation program. This capability must be preserved during the transition from military standards to commercial practices.

4. Fourth Subsidiary Question

How will this change in policy affect potential government contractors for armament system software?

Most current and potential government software contractors indicated they will take a "wait-and-see" position on the policy change. Current government contractors will continue to work on DoD software projects. Prospective government contractors will react based upon their assessment of DoD actions implementing the policy change.

The following actions are recommended to reduce the impact of the change from military standards to commercial standards:

- Educate and train government software professionals and technicians on existing industrial standards for software development and maintenance.
- Provide early and continuous communication in professional software publications of implementation guidance.
- Ensure DoD and commercial software developers equally participate in development of the new standard.

Contractors are a critical part of the embedded armament software community. Communications are crucial to maintaining good government/contractor relations.

5. Fifth Subsidiary Question

Will this change in policy affect risk in the development and maintenance of armament systems?

The answer is certainly yes. However, the extent of the change in risk cannot be determined at this time. Additional experience with the new policy is required to gather sufficient data so that potential risk factors can be identified and their consequences and probabilities of occurrence evaluated.

The following actions are recommended to reduce the potential risks resulting from the transition to commercial standards:

- Tailor and use MIL-STD-498.
- Encourage substitution of properly tailored commercial standards for individual MIL-STD-498 requirements, where applicable, while the new commercial software standard is being developed.
- Educate and train government software professionals on the tailoring and use of existing commercial standards and specifications.
- Ensure government participation in the effort by IEEE/EIA to develop a new commercial software standard to replace MIL-STD-498.
- Provide early and continuous communication in government and civilian software and acquisition professional publications and forums of information about experience with the policy change and policy implementation guidance.
- Ensure that the commercial software standard replacing MIL-STD-498 is completed and approved within the two year transition period.
- Ensure that the necessary changes to the DFARS and other DoD regulations and instructions are completed within the specified transition period.

While some risk will undoubtedly result from the change to commercial software practices, the effect of the risk can be mitigated through actions now. Government and industry should form a partnership to identify and control potential risk in military software development and maintenance.

6. Primary Research Question

What will be the effect on armament system acquisitions of allowing commercial practices to be used instead of requiring DoD or military standards in the development and maintenance of embedded software?

The potential risks in embedded armament system software development and maintenance will change. Government and civilian software professionals will be required to exercise special care to identify, assess and control risk in embedded software development and maintenance. During a two year transition period, a new comprehensive commercial software standard will be developed to replace MIL-STD-498. IEEE and EIA are jointly developing the new standard. This will also allow DoD to develop guidelines and training to implement the new standard.

The following actions are recommended to reduce the impact and potential risks resulting from the transition to commercial standards:

- Tailor and use MIL-STD-498.
- Encourage substitution of properly tailored commercial standards for individual MIL-STD-498 requirements, where applicable, while the new commercial software standard is being developed.
- Educate and train government software professionals on the tailoring and use of existing commercial standards and specifications immediately.
- Ensure government participation in the effort by IEEE/EIA to develop a new commercial software standard to replace MIL-STD-498. The government participants should include representatives from the embedded armament software management, maintenance and test and evaluation disciplines.
- Provide early and continuous communication in government and civilian software and acquisition professional publications and forums of information about experience with the policy change and policy implementation guidance.
- Ensure that the commercial software standard replacing MIL-STD-498 is completed and approved within the two year transition period.

- Ensure that the necessary changes to the DFARS and other DoD regulations and instructions are completed within the specified transition period.

Government and industry should form a partnership to develop the new commercial software standard. That partnership should not end with development of the new standard. Both government and industry need to educate their software professionals on tailoring and application of the new standard. During the transition period, the keys to success are cooperation, education and communication.

B. RECOMMENDATIONS FOR FURTHER RESEARCH

This section provides a list of topics identified during this investigation of the thesis questions as requiring additional research.

1. Commercial Practices in Software Development and Maintenance

Examine the impact of transitioning from military standards to commercial practices on software development for non-armament systems and Automated Information Systems (AIS). This research effort looked only at the impact of transitioning to commercial practices on Army armament systems. The Army armament community's software requirements and background are different from other DoD software communities. Other software communities may have a vastly different reaction to the policy change. Additionally, this research effort was only able to assess the initial response to the change to commercial standards. The policy change was directed on 29 June 1994. MIL-STD-498 was approved on 8 November 1994 and was not published until 5 December 1994. This research on the topic was terminated in late February 1995 in order to complete the thesis within time constraints. As the military software community gains experience with using commercial standards, their knowledge of the impacts and risks may change. An understanding of the impacts and risks associated with the policy change can provide a basis for "fine tuning" implementation guidelines.

2. Risks Associated With Using Commercial Practices to Develop Military Software

Examine the specific risks associated with using existing commercial practices to develop military software. Identify and assess the difference in risk between using military standards and commercial standards to develop military embedded software or AIS. The study should develop alternative recommendations to reduce the risks identified. The results of this study clearly indicate an incomplete understanding of the risks involved in changing from military standards to commercial software practices. The first step in controlling risk is identification. Controlling risk is vital to the successful development of military software.

3. IEEE/EIA Standard XXXX

Examine the new commercial software standard being developed to replace MIL-STD-498. The study could investigate the development process for the new standard. The study should identify linkages between the new standard and MIL-STD-498 and existing commercial software standards. The study may investigate tailoring of the new standard to meet specific military requirements. The study should identify potential benefits and problem areas associated with the new standard. Since this standard will be used in the future to develop the military's software, a thorough understanding of the new standard will be required to ensure its efficient and effective use.

4. Cost Reduction and Commercial Software Standards

Examine the potential of the transition from military standards to commercial standards to effect reductions in the cost of software. One of the goals of the change to commercial standards was to reduce the cost of military systems. The study should identify specific measures that could result in savings in the development and maintenance of military software without jeopardizing performance or supportability. Reducing the cost of

software development and maintenance would free up scarce funds to support other priority activities.

5. Commercial Practices in the Acquisition of Military Hardware

Examine the impact of transitioning from DoD and military standards and specifications to commercial standards and specifications on military hardware acquisition. This research effort looked only at the impact of the change on embedded armament system software. Embedded armament system software is only a small fraction of the defense research, development and acquisition budget. Early insight into the impact of transitioning to commercial practices on hardware development and acquisition can provide feedback to the policy makers and be used to adjust implementation guidelines.

C. A FINAL THOUGHT

This study provides initial insight into the beginning of the transition from military standards to commercial practices for embedded armament system software development and maintenance. The research assessed the early impact of the policy change on contracting for development and maintenance, test and evaluation maintenance, and contractors of military software. The study makes recommendations to reduce the impact resulting from this change. Additionally, it identified potential risks associated with this change and makes recommendations to mitigate their effects. This study provides a basis for managing the transition to commercial practices and refining policy implementing the change. This study should not be viewed as the end of research on this topic, but as the starting point for further study into the new commercial software standard and implementing guidance as it develops.

LIST OF REFERENCES

1. Gagnon, Carrol, Deputy Product Manager Paladin, Interview, January 1995.

2. Perry, William J., Secretary of Defense, *Specifications and Standards -- A New Way of Doing Business*, Memorandum, 29 June 1994.

3. Department of Defense, DOD-STD-2167A, *Military Standard Defense System Software Development*, 29 February 1988.

4. Bergman II, Walter B., Assistant Secretary of Defense for Economic Security, *MIL-STD-498 "Software Development and Documentation"*, Memorandum, 8 November 1994.

5. Defense Systems Management College, *Mission Critical Computer Resources Management Guide*, 1989.

6. Boehm, Barry W., "A Spiral Model of Software Development and Enhancement", *Software Management*, Donald J. Reifer, editor, IEEE Computer Society Press, 1993.

7. Heil, James H., *Overview of MIL-STD-498*, MITRE Briefing, 30 January 1995.

8. Department of Defense, MIL-STD-498, *System Software Development and Documentation*, 5 December 1995.

9. United States Army, Historical files of the Office of the Assistant Deputy Chief of Staff for Operations and Plans for Force Development (DAMO-FDG).

10. United States Army, *Required Operational Capability (ROC) for the Howitzer Improvement Program*, July 1985.

11. United States Army, Field Manual (FM) 6-140, *The Field Artillery Battery*, February 1988.

12. United States Army, *Howitzer Improvement Program Operational and Organizational Plan (O&O Plan)*, January 1985.

13. Product Manager Paladin, *Howitzer Improvement Program Fact Book*, March 1990.

14. Delcoco, Gene, Deputy Program Manager Crusader, Interview, January 1995.

15. Yung, Larry, Chief of Command, Control, Communications and Computers for Program Manager Crusader, Interview, February 1995.

16. Batchis, George, Deputy Program Manager SADARM, Interview, January 1995.

17. Beck, Peter, Chief of Software for Program Manager SADARM, Interview, February 1995.

18. McCaffrey, Martin J., "Maintenance of Expert Systems -- The Upcoming Challenge", *Managing Expert Systems*, Idea Group Publishing, 1992.

19. National Bureau of Standards, *Report on Software*, Efriam Turbon, editor, National Bureau of Standards, November 1988.

20. Kitfield, James, "Is Software DoD's Achilles' Heel?", *Military Forum*, July 1989.

21. Wahl, Steven J., System Engineer for Product Manager Paladin, Interview, February 1995.

22. Adams, Dale G., Program Executive Officer for Field Artillery Systems, Interview, December 1994.

23. Defense Systems Management College, *Risk Management, Concepts and Guidance*, 1994.

24. Jones, Capers. *Assessment and Control of Software Risks*, Yourdon Press, 1994.

25. Boehm, Barry W., "Software Risk Management: Principles and Practices", *Software Management*, Donald J, Reifer, editor, IEEE Computer Society Press, 1993.

26. Irby, Dewitt, T. Jr., Major General, United States Army, Presentation at Naval Postgraduate School, 16 February 1995.

LIST OF INTERVIEWS

1. Adams, Dale G., Program Executive Officer for Field Artillery Systems, Picatinny Arsenal, NJ, December 1994.

2. Batchis, George, Deputy Program Manager SADARM, Picatinny Arsenal, NJ, January 1995.

3. Beck, Peter, Chief of Software for Program Manager SADARM, Picatinny Arsenal, NJ, February 1995.

4. Casper, John, Chief of Software Development for Product Manager Paladin, Picatinny Arsenal, NJ, February 1995.

5. Delcoco, Gene, Deputy Program Manager Crusader, Picatinny Arsenal, NJ, January 1995.

6. Devine, Michael P., Associate Program Executive Officer for Field Artillery Systems for Integration, Picatinny Arsenal, NJ, January 1995.

7. Gagnon, Carol, Deputy Product Manager Paladin, Picatinny Arsenal, NJ, January 1995.

8. Griffin, Darold, Deputy to the Commander of the Army Materiel Command, Alexandria, VA, December, 1994.

9. Heil, James H., Member of Technical Staff, MITRE Corporation, Eatontown NJ, February 1995.

10. Huangfu, Austin, Chief of Command, Control, Communications and Intelligence Systems, Director of Operational Test and Evaluation, Pentagon, Washington, DC, January 1995.

11. Nathan, Daniel, Chief, Life Cycle System Software Center, Picatinny Arsenal, NJ, February 1995.

12. Sierodzonski, Joseph, System Engineer, Fire Control and Software Engineering Branch, Fire Support and Armaments Center, Picatinny Arsenal, NJ, January 1995.

13. Wahl, Steven J., System Engineer for Product Manager Paladin, Picatinny Arsenal, NJ, February 1995.

14. Yung, Larry, Chief of Command, Control, Communications and Computers for Program Manager Crusader, Picatinny Arsenal, NJ, February 1995.

15. Program Manager, Company A, Large Defense Oriented Corporation (hardware and software oriented), February 1995.

16. Program Manager, Company B, Large Defense Oriented Corporation (software oriented), February 1995.

17. Program Manager, Company C, Large Defense Oriented Corporation (munitions and software oriented), February 1995.

18. Program Manager, Company D, Large Defense Oriented Corporation (communications, electronics and software oriented), February 1995.

19. Project Manager, Company E, Medium Defense Oriented Corporation (software maintenance oriented), February 1995.

20. Vice President for Software, Company F, Small Commercial and Defense Corporation (software development and maintenance oriented), January 1995.

21. Division Chief, Company G, Large Engineering Corporation, (aerospace, electronics and software engineering oriented), January 1995.

NOTE: Company A through G representatives did not want their names or the names of their companies cited.

BIBLIOGRAPHY

Armavas, Donald P. and William J. Ruberry, *Government Contract Guidebook*, Federal Publications Inc., 1992.

Bergman II, Walter B., Assistant Secretary of Defense for Economic Security, *MIL-STD-498 "Software Development and Documentation"*, Memorandum, 8 November 1994.

Boehm, Barry W., "A Spiral Model of Software Development and Enhancement", *Software Management*, Donald J. Reifer, editor, IEEE Computer Society Press, CA, 1993.

Boehm, Barry W., "Software Risk Management: Principles and Practices", *Software Management*, Donald J. Reifer, editor, IEEE Computer Society Press, 1993.

Branscom, Lewis and George Parker. "Funding Civilian and Dual-Use Industrial Technology", *Empowering Technology, Implementing a US Strategy*, MIT Press, 1993.

Defense Systems Management College, *Acquisition Strategy Guide*, 1984.

Defense Systems Management College, *Commercial Practices for Defense Acquisition Guidebook*, January 1992.

Defense Systems Management College, *Mission Critical Computer Resources Management Guide*, 1989.

Defense Systems Management College, *Risk Management, Concepts and Guidance*, 1994.

Department of Defense Directive 5000.1, *Major and Non-Major Defense Acquisition Programs*, 23 February 1991.

Department of Defense Directive 5000.2, *Defense Acquisition Procedures*, 23 February 1991.

Department of Defense Directive 5000.29, *Management of Computer Resources in Major Defense Systems*, 28 December 1976.

Department of Defense Directive 5000.3, *Test and Evaluation*, 28 April 1989.

Department of Defense Instruction 5000.2, *Defense Acquisition Management Policies and Procedures*, 23 February 1991.

Department of Defense Manual 5000.2-M, *Defense Acquisition Management Documentation and Reports*, February 1991.

Department of Defense, *DoD Supplement to the Federal Acquisition Regulation*, 1988.

Department of Defense, DOD-STD-1467, *Software Support Environment*, 18 January 1985.

Department of Defense, DOD-STD-1679, *Defense System Software Development*, July 1982.

Department of Defense, DOD-STD-2167A, *Military Standard Defense System Software Development*, 29 February 1988.

Department of Defense, DOD-STD-2168, *Defense System Software Quality Program*, 29 April 1988.

Department of Defense, MIL-STD-498, *System Software Development and Documentation*, 5 December 1994.

Department of Defense, MIL-STD-499B, *Systems Engineering*, 6 May 1992.

Department of Defense, Military Handbook, MIL-HDBK-286, *Software Quality Evaluation*, 31 December 1985.

Department of Defense, Military Handbook, MIL-HDBK-287, *A Tailoring Guide for DOD-STD-2167A, Defense Systems Software Development*, 11 August 1989.

Department of Defense, Military Handbook, MIL-HDBK-782, *Software Support Environment Acquisition*, 29 February 1989.

Downward, Bernadett Gagnon, "A Brave New World: Melding Systems and Software Engineering", *Systems Engineering a Competetive Edge in a Changing World*, Proceedings of the Fourth Annual International Symposium of the National Council on Systems Engineering, August 10 - 12, 1994.

Glass, Robert L., *Software Reliability Guidebook*, Englewook Cliffs; Prentice-Hall, 1979.

Heil, James H., *Overview of MIL-STD-498*, MITRE Briefing, 30 January 1995.

Irby, Dewitt, T. Jr., Major General, United States Army, Presentation at Naval Postgraduate School, 16 February 1995.

Humphrey, Watts S., *Managing the Software Process*, Addison-Wesley Publishing Company, 1989.

Institute for Defense Analysis. *The Future of Military R & D: Towards a Flexible Acquisition Strategy*, IDA Paper P-2444, Alexandria VA, Defense Technical Information Center, 1990.

Jones, Anita, Director, Defense Research and Engineering, *Statement to the Subcommittee on Research and Technology of the House Committee on Armed Services*, 14 April, 1994.

Jones, Capers., *Assessment and Control of Software Risks*, Yourdon Press, 1994.
Kitfield, James, "Is Software DoD's Achilles' Heel?", *Military Forum*, July 1989.

McCaffrey, Martin J., "Maintenance of Expert Systems -- The Upcoming Challenge", *Managing Expert Systems*, Idea Group Publishing, 1992.

National Bureau of Standards, *Report on Software*, Efrain Turbon, editor, National Bureau of Standards, November 1988.

O'Brien, James A, *Management Information Systems, A Managerial End User Perspective*, Richard D. Irwin Inc., 1993.

Perry, William J., Secretary of Defense, *Specifications and Standards -- A New Way of Doing Business*, Memorandum, 29 June 1994.

Process Action Team on DoD Military Specifications and Standards, *Blueprint for Change*, Office of the Deputy Under Secretary of Defense for Acquisition Reform, April 1994.

Product Manager Paladin, *Howitzer Improvement Program Fact Book*, March 1990.

Reifer, Donald J., *Software Management*, IEEE Computer Society Press, 1993.

Rheinmetall, *Handbook on Weaponry*, Rheinmetall GmbH, 1982.

U. S. Congress, Office of Technology Assessment., "Research and Development", *In Building Future Security*, OTA-ISC-530. Washington, DC USGPO, June 1992.

United States Army, Field Manual, FM 6-140, *The Artillery Battery*, February 1988.

United States Army, *Howitzer Improvement Program Operational and Organizational Plan (O&O Plan)*, January 1985.

United States Army, *Required Operational Capability (ROC) for the Howitzer Improvement Program*, July 1985.

United States Government, *Federal Acquisition Regulation (FAR)*, 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5100	2
3.	Defense Logistic Studies Exchange U.S. Army Logistics Management College Fort Lee, Virginia 23801-6043	1
4.	Professor David V. Lamm (Code SM/LT) Naval Postgraduate School Monterey, California 93943-5100	4
5.	Professor Martin J. McCaffrey (Code SM/MF) Naval Postgraduate School Monterey, California 93943-5100	6
6.	Professor Orin E. Marvel (Code SE/OM) Naval Postgraduate School Monterey, California 93943-5100	2
7.	LTC John T. Dillard (Code SM/JD) Naval Postgraduate School Monterey, California 93943-5100	1
8.	OASA (RDA) ATTN: SARD-ZAC 103 Army Pentagon Washington, D.C. 20310-0103	1
9.	Program Executive Officer, Field Artillery Systems ATTN: SFAE-FAS (Mr. Dale Adams) Picatinny Arsenal, New Jersey 07806-5000	2

10. Program Manager, Crusader 1
ATTN: SFAE-FAS-AFAS (Mr. Gene Delcoco)
Picatinny Arsenal, New Jersey 07806-5000

11. Program Manager, Crusader 1
ATTN: SFAE-FAS-AFAS (Mr. Larry Yung)
Picatinny Arsenal, New Jersey 07806-5000

12. Program Manager, Paladin 1
ATTN: SFAE-FAS-HIP (Mr. Carrol Gagnon)
Picatinny Arsenal, New Jersey 07806-5000

13. Program Manager, Paladin 1
ATTN: SFAE-FAS-HIP (Mr. John Casper)
Picatinny Arsenal, New Jersey 07806-5000

14. Program Manager, Paladin 1
ATTN: SFAE-FAS-HIP (Mr. Steven Wahl)
Picatinny Arsenal, New Jersey 07806-5000

15. Program Manager, Sense and Destroy Armor (SADARM) 1
ATTN: SFAE-FAS-SD (Mr. George Batchis)
Picatinny Arsenal, New Jersey 07806-5000

16. Program Manager, Sense and Destroy Armor (SADARM) 1
ATTN: SFAE-FAS-SD (Mr. Peter Beck)
Picatinny Arsenal, New Jersey 07806-5000

17. Life Cycle Software Support Center 1
ATTN: Chief (Mr. Dan Nathan)
Picatinny Arsenal, New Jersey 07809-5000

18. Fire Support and Armaments Center 1
Fire Control and Software Engineering Branch
ATTN: AMSTA-AR-FSF-B (Mr. Joseph Sierodzonski)
Picatinny Arsenal, New Jersey 07809-5000

19. Director of Operational Test and Evaluation 1
Chief C3I Systems (Mr. Austin Huangfu)
Pentagon
Washington, D.C. 20310

20. Mr. James H. Heil 1
MITRE Corp. -- J83
145 Wyckoff Road
Eatontown, New Jersey 07724

21. Thomas E. Mullins 2
7511 Tendring Trail
Manassas, Virginia 22111